/ 006447 /

Am2969

Memory Timing Controller with EDC Timing Control

PRELIMINARY

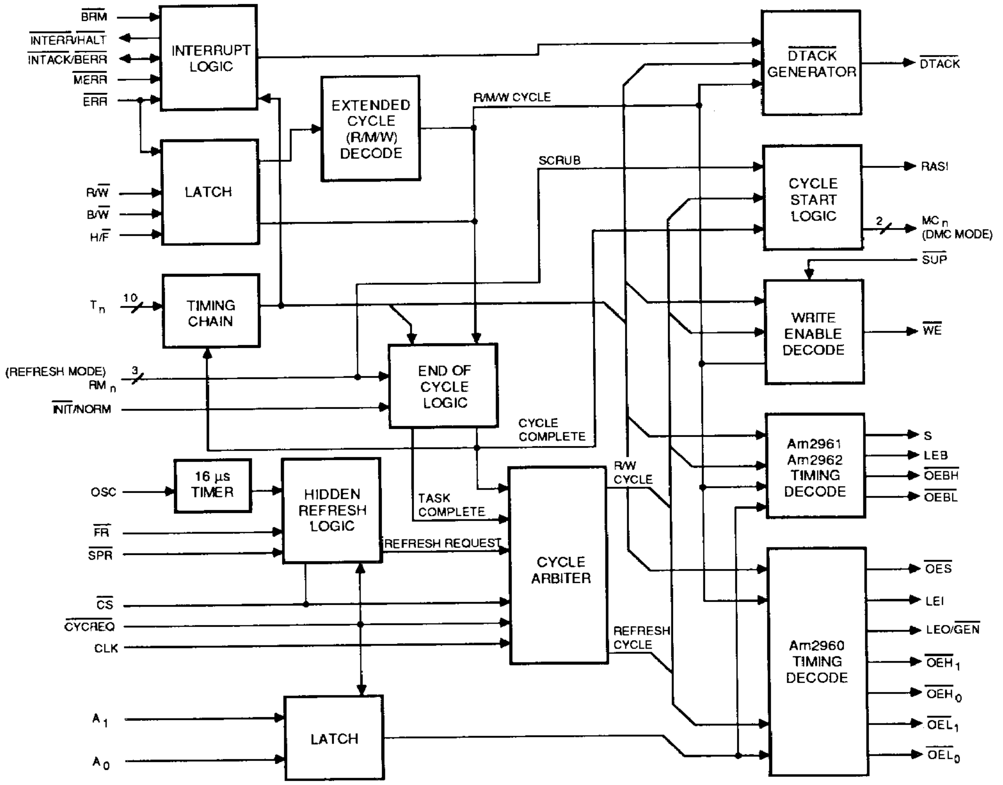
Am2969

Advanced Micro Devices

DISTINCTIVE CHARACTERISTICS

- Provides complete timing control for memory systems using 64K, 256K, 1M, or 4M DRAMs and AMD's Dynamic Memory Support chip set
- Interfaces directly with all 16-bit processors and supports most 32-bit processors
- Performs memory initialization (DRAMs up to 1M, including 1M x 4)
- WE output directly drives up to 88 DRAMs
- Arbitrates between refresh requests and memory access requests
- Refresh support: 128, 256, or 512 cycle burst refresh, distributed, scrubbing, automatic, or hidden modes
- Refresh control: internal or external
- On-chip interval timer for automatic refresh
- Synchronous or asynchronous arbitration of memory cycles
- Supports bus-retry
- On-chip interrupt logic
- Supports word, long-word, and byte operations
- Controls AMD's Memory Support chip set to perform AMD's patented Scrubbing Refresh/EDC cycle

BLOCK DIAGRAM



05221C-1

3 R orig

Publication #	Rev.	Amendment
05221	C	/0

Issue Date: September 1987

006446

GENERAL DESCRIPTION

The Am2969 is a high-performance enhanced memory timing controller (MTC) that provides the timing control in AMD's Dynamic Memory Support chip set. The Am2969 MTC replaces much of the "glue" logic necessary in DRAM systems by generating all control signals for the Am2968A/Am29368 Dynamic Memory Controllers, Am2961/Am2962 Multiple EDC Bus Buffers, and the Am2960/Am2960A Error Detection and Correction Units. It manages Dynamic RAM access and refresh, processor interface, and error detection and correction (EDC) processes for the CPU. The chip can be configured for word, long-word, and byte operations (including the data controls for byte write with error correction).

The Am2969 coordinates the timing and control signals needed for a processor and memory to function interactively as a meaningful part of a computer system. The MTC chip takes advantage of AMD's "building block" approach by drawing on the Am2971 Programmable Event Generator (PEG™) chip as its timing reference. By using the PEG (or a delay line) the user can tailor the system to provide precise timing, maximize system performance, and achieve greater design flexibility.

The Am2969 also provides an internal refresh interval timer to generate refresh requests independent of the CPU (distributed refreshes). This frees up valuable CPU time and makes the Dynamic RAM appear static. It also guarantees proper refresh timing under all combinations of CPU and DMA requests.

Additionally, the Am2969 acts as the arbiter between memory refresh requests and CPU Read/Write requests. Arbitration can be for either synchronous or asynchronous memory cycles. The chip also performs memory initialization, controls various timing and data flows, and handshakes with the CPU.

Other operations of the Am2969 include bus-retry, on-chip interrupt logic, and facilities for interfacing with a wide range of microprocessors. A single Am2969 can support the timing requirements for all 16-bit systems and most 32-bit systems.

RELATED AMD PRODUCTS

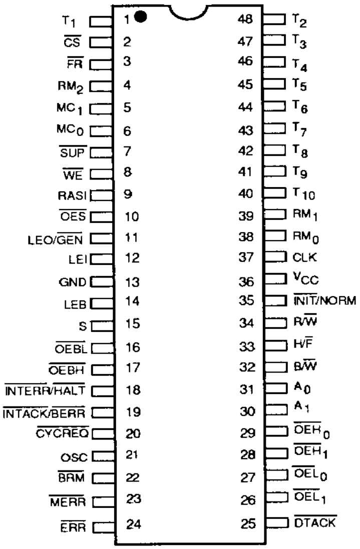
Part No.	Description
Am29368	Dynamic Memory Controller for 1M DRAMs
Am2960/A	16-Bit Error Detection and Correction Units
Am29C60/A	CMOS 16-Bit Error Detection and Correction Units
Am2961	Inverting EDC Multiple Bus Buffer
Am2962	Non-Inverting EDC Multiple Bus Buffer
Am2968A	Dynamic Memory Controller for 256K DRAMs
Am2970	Memory Timing Controller (without EDC Control)
Am2971/71A	Programmable Event Generators (PEG)
Am29827/28	Non-Inverting High-Performance Three-State Buffers
Am29845	Inverting High-Performance Octal Bus Interface Latch
Am29846	Non-Inverting High-Performance Octal Bus Interface Latch

PEG is a trademark of Advanced Micro Devices, Inc.

CONNECTION DIAGRAMS

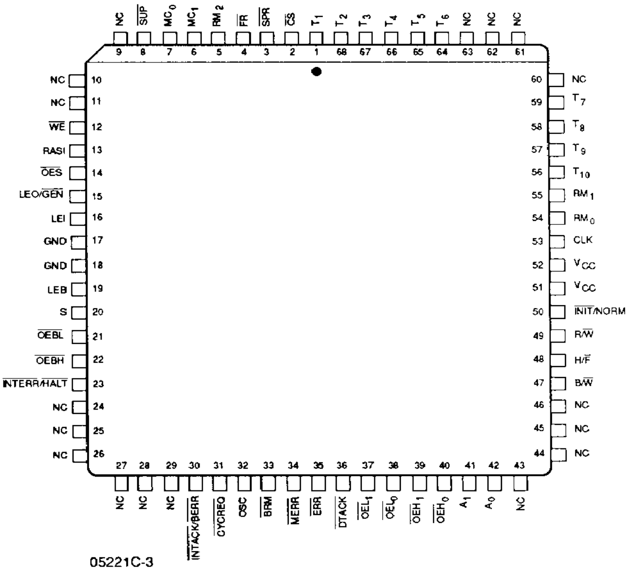
Top View

DIPs*



05221C-2

PLCC**

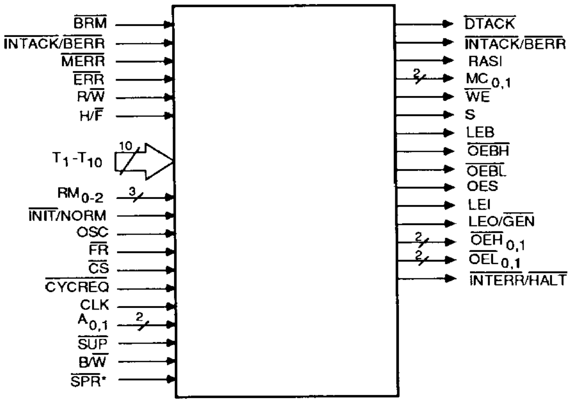


05221C-3

Note: Pin 1 is marked for orientation.

* Also available in 48-pin Flatpack; pinout is identical to DIPs.
** Also available in 68-pin LCC; pinout is identical to PLCC.

LOGIC SYMBOL



05221C-4

*The function $\overline{\text{SPR}}$ is available as a pin on the surface-mount packages only. To implement this function in other packages, refer to the Functional Description.

Package and Chip Information

Die Size: 0.199" x 0.153"   Gate Count: 450

Parameter	Ceramic DIP	Plastic DIP	Ceramic LCC	Plastic LCC	Ceramic Flatpack	Unit
θ_{JA}	37	55	31	35	–	°C/W
θ_{JC}	10	NA	6	NA	–	°C/W

ORDERING INFORMATION

Standard Products

AMD standard products are available in several packages and operating ranges. The order number (Valid Combination) is formed by a combination of: **a. Device Number**
b. Speed Option (if applicable)
c. Package Type
d. Temperature Range
e. Optional Processing

AM2969 D C B

e. OPTIONAL PROCESSING
Blank = Standard processing
B = Burn-in

d. TEMPERATURE RANGE
C = Commercial (0 to +70°C)
E = Extended Commercial (−55 to +125 C)

c. PACKAGE TYPE
P = 48-Pin Plastic DIP (PD 048)
D = 48-Pin Sidebrazed Ceramic DIP (CD 048)
J = 68-Pin Plastic Leaded Chip Carrier (PL 068)
L = 68-Pin Ceramic Leadless Chip Carrier (CL 068)
X = Dice

b. SPEED OPTION
Not Applicable

a. DEVICE NUMBER/DESCRIPTION
Am2969
Memory Timing Control with EDC Timing Control

Valid Combinations	
AM2969	PC, PCB, DC, DCB, JC, JCB, LC, LCB, XC, DE, DEB

Valid Combinations

Valid Combinations list configurations planned to be supported in volume for this device. Consult the local AMD sales office to confirm availability of specific valid combinations, to check on newly released combinations, and to obtain additional data on AMD's standard military grade products.

MILITARY ORDERING INFORMATION

APL Products

AMD products for Aerospace and Defense applications are available in several packages and operating ranges. APL (Approved Products List) products are fully compliant with MIL-STD-883C requirements. The order number (Valid Combination) for APL products is formed by a combination of: **a. Device Number**
b. Speed Option (if applicable)
c. Device Class
d. Package Type
e. Lead Finish

AM2969 /B X C

e. LEAD FINISH
A = Hot Solder Dip
C = Gold

d. PACKAGE TYPE
X = 48-Pin Sidebrazed Ceramic DIP (SD 048)
Y = 48-Pin Ceramic Flatpack (CFT048)
U = 68-Pin Ceramic Leadless Chip Carrier (CL 068)

c. DEVICE CLASS
/B= Class B

b. SPEED OPTION
Not Applicable

a. DEVICE NUMBER/DESCRIPTION
Am2969
Memory Timing Controller with EDC Timing Control

Valid Combinations	
AM2969	/BXC, /BYC, /BUA

Valid Combinations

Valid Combinations list configurations planned to be supported in volume for this device. Consult the local AMD sales office to confirm availability of specific valid combinations or to check on newly released valid combinations.

Group A Tests

Group A tests consist of subgroups 1, 2, 3, 7, 8, 9, 10, 11.

PIN DESCRIPTION

$A_{0,1}$ Address Bits 0 and 1 (Inputs)

These are the two least significant bits of the CPU address. They are used to control byte and 16-bit word and 32-bit long-word operations. A_0 and A_1 are latched on the HIGH-to-LOW transition of $\overline{\text{CYCREQ}}$.

$\overline{\text{BRM}}$ Bus Retry Mode (Input; Active LOW)

When this input is LOW, the Am2969 enters the Bus Retry Mode. In this mode, the $\overline{\text{INTERR}}$ and $\overline{\text{INTACK}}$ pins assume the $\overline{\text{HALT}}$ and $\overline{\text{BERR}}$ functions, respectively. This input should be tied HIGH if the processor does not support bus retry.

B/$\overline{\text{W}}$ Byte/$\overline{\text{Word}}$ (Input)

This input indicates a byte write operation is to be performed when HIGH and a word write operation when LOW. This signal is a "Don't Care" during read cycles.

CLK Clock (Input)

For systems requiring synchronous arbitration of memory access and refresh requests, this input receives the system clock. The falling edge of CLK causes RASI to go HIGH and $MC_{0,1}$ to switch. For asynchronous arbitration, this input must be tied HIGH.

$\overline{\text{CS}}$ Chip Select (Input; Active LOW)

When $\overline{\text{CS}}$ is LOW, the MTC is enabled. A memory read/write cycle can only be performed when $\overline{\text{CS}}$ is active. Refresh cycles occur independent of $\overline{\text{CS}}$. When $\overline{\text{CS}}$ is HIGH, all memory requests (HIGH-to-LOW transitions of $\overline{\text{CYCREQ}}$) will be interpreted as refresh requests ("Hidden" refresh).

$\overline{\text{CYCREQ}}$ Cycle Request (Input; Active LOW)

When $\overline{\text{CS}}$ is LOW, this input will generate an internal memory request for the Am2969 on a HIGH-to-LOW transition. When $\overline{\text{CS}}$ is HIGH, a HIGH-to-LOW transition of $\overline{\text{CYCREQ}}$ will generate a hidden refresh request.

$\overline{\text{DTACK}}$ Data Transfer Acknowledge (Output; Open Collector, Active LOW)

The HIGH-to-LOW transition of $\overline{\text{DTACK}}$ informs the CPU that a write cycle has begun, or that data will be on the system bus at the correct time during a read cycle. The time at which $\overline{\text{DTACK}}$ occurs is set at T_1 or T_4.

$\overline{\text{ERR}}$ Error (Input; Active LOW)

When this signal goes LOW, it indicates that an error has occurred on a read cycle. This input should be connected to the $\overline{\text{ERR}}$ output of the EDC unit. This input is sampled at the rising edge of T_2.

$\overline{\text{FR}}$ Forced Refresh (Input; Active LOW)

This input is used to force a refresh cycle at user-designated times. The falling edge of $\overline{\text{FR}}$ latches an internal refresh request. If the memory is busy, the refresh is done at the completion of the current cycle.

GND +0-V Ground Reference

Refer to the connection diagrams for placement and number of GND pins. Refer to the Functional Description for information on proper decoupling.

H/$\overline{\text{F}}$ Half/$\overline{\text{Full}}$ (Input)

This input indicates the relative size of the system bus with respect to the memory bus. It is HIGH if the system bus is half the memory bus width, and LOW if the system bus width is equal to the memory bus width.

$\overline{\text{INTACK}}$/$\overline{\text{BERR}}$ Interrupt Acknowledge/Bus Error (Input; Active LOW/Output; Active LOW, Open Collector)

When $\overline{\text{INTACK}}$ goes LOW, it will reset the Interrupt Request ($\overline{\text{INTERR}}$) output on the Am2969. In the Bus Retry Mode ($\overline{\text{BRM}}$ = LOW), this pin will act as the BUS ERROR output.

$\overline{\text{INTERR}}$/$\overline{\text{HALT}}$ Interrupt Error/Halt (Output; Active LOW, Open Collector/Output; Active LOW, Open Collector)

When $\overline{\text{BRM}}$ = HIGH, this output interrupts the processor if a single-bit error is detected in the Fly-By Mode or if a multiple-bit error is detected in the Flow-Thru Mode. **CAUTION:** If T_4 occurs before $\overline{\text{INTACK}}$, $\overline{\text{INTERR}}$ will be reset by T_4. When $\overline{\text{BRM}}$ = LOW, this pin is defined as $\overline{\text{HALT}}$. This is called Bus-Retry Mode, and the Am2969 will request the last cycle to be rerun by asserting a $\overline{\text{BERR}}$ and $\overline{\text{HALT}}$ simultaneously if a single-bit error occurs. If a multiple-bit error occurs, only $\overline{\text{BERR}}$ will be asserted.

$\overline{\text{INIT}}$/NORM Inititalize/Normal (Input)

When the Am2969 detects a falling edge on this input, it will enter the initialize mode of operation. This allows the memory to be initialized easily with a minimum of CPU overhead. When this input is HIGH, normal Read/Write and refresh cycles may be performed.

LEB Latch Enable Buffer (Output; Active HIGH)

This output is intended to be connected to the LEB input of the Am2961/Am2962 EDC Bus Buffers. It controls the latching of data going from the EDC Bus Buffers to the system bus. When LEB is HIGH, the latch is transparent; when LOW, the data is latched.

LEI Latch Enable In (Output; Active HIGH)

This output should be connected to the LEI input of the Am2960/Am2960A EDC Unit. It controls the latching of data (data and check bits) into the EDC unit. The latch is transparent when LEI is HIGH, and the data is latched in when LEI is LOW.

LEO/$\overline{\text{GEN}}$ Latch Enable Out/$\overline{\text{Generate}}$ (Output)

This output is intended to be connected to the LEO and $\overline{\text{GEN}}$ inputs of the Am2960/Am2960A EDC Unit. LEO OUT controls the latching of corrected data from the EDC unit into its output latch. $\overline{\text{GEN}}$ instructs the Am2960/Am2960A to generate check bits for the data in its input latch. When LEO/$\overline{\text{GEN}}$ is HIGH, the latch is transparent; when LOW, data is latched and the EDC unit goes into the Check Bit Generation Mode.

$\text{MC}_{0,1}$ Mode Control Bits 0 and 1 (Outputs; Active HIGH)

These outputs are connected to the $\text{MC}_{0,1}$ inputs of the Am2968A/Am29368 DMC. They indicate the type of memory cycle to be performed. Possible combinations and mode descriptions appear in the Functional Description in Table 1.

$\overline{\text{MERR}}$ Multiple Error (Input; Active LOW)

This input identifies that a multiple error has been detected by the EDC unit. This input should be connected to the $\overline{\text{MERR}}$ (multiple error) output of the Am2960/Am2960A.

$\overline{\text{OEBH}}$, $\overline{\text{OEBL}}$ Output Enable Bus HIGH/LOW (Outputs; Active LOW)

These outputs are used when interfacing to a system bus, which can be either the same width as memory or half the width. They control the multiplexing of half-words onto the system bus from the EDC Bus Buffers. $\overline{\text{OEBH}}$ and $\overline{\text{OEBL}}$ are enabled as shown in Table 5 of the Functional Description. These outputs are to be connected to the appropriate $\overline{\text{OEB}}$ inputs of the Am2961/Am2962. When they are HIGH, the system bus outputs of the Am2961/Am2962 are in the high-impedance state. When LOW, the Am2961/Am2962 system bus outputs are enabled.

$\overline{\text{OEH}}_{0,1}$ $\overline{\text{OEL}}_{0,1}$ Output Enable HIGH/LOW (Outputs; Active LOW)

These signals are used to select the appropriate output(s) from the EDC Bus Buffers and EDC Unit(s) on the Y-bus, when performing either byte writes or word writes when H/$\overline{\text{F}}$ is HIGH. $A_{0,1}$ are decoded so that the selected byte or word may be written into memory. These signals are intended to be connected to the corresponding $\overline{\text{OE}}$ Byte inputs of the Am2960/Am2960A and OEY inputs of the Am2961/Am2962. Table 3 of the Functional Description shows how these outputs are enabled.

$\overline{\text{OES}}$ Output Enable Syndromes (Output; Active LOW)

This output is intended to be connected to the $\overline{\text{OE}}$ SC input of the Am2960/Am2960A EDC Unit. It controls the feedback of the syndrome bits in the 32-bit mode.

OSC Oscillator (Input)

The oscillator input signal is used to generate an internal refresh clock. It is this oscillator which initiates a refresh cycle if $\overline{\text{FR}}$ does not go active. The OSC signal comes from external components (illustrated in Figure 5 of the Functional Description).

RASI Row Address Strobe Input (Output; Active HIGH)

This output is connected to the RASI input of the Am2968A/Am29368 DMC. It is used to start a memory access for the DMC. The RASI output is also connected to the TRIG input of the PEG or to a delay line to start the timing sequence. The rising edge of RASI initiates the memory access for the DMC, and the leading edge initiates the PEG (or delay line).

RM_{0-2} Refresh Mode (Inputs)

These inputs control the type of refresh cycle that the Am2969 will initiate. Table 4 of the Functional Description provides the valid input combinations and the resulting refresh mode selection.

R/$\overline{\text{W}}$ Read/Write (Input)

This input indicates a memory read request when HIGH, and a write request when LOW.

S Select (Output)

The S output should be connected to the S input of the Am2961/Am2962 EDC Bus Buffers. It controls the source of data for the Am2960/Am2960A EDC Unit. When HIGH, data comes from the system bus; when LOW, data comes from the memory. Refer to Figure 3 of the Functional Description for a detailed description of the interface between the Am2969 and the Am2961/Am2962 in the Fly-By mode.

$\overline{\text{SPR}}$ System Power-up Reset (Input; ActiveLOW)

$\overline{\text{SPR}}$ is an active-LOW signal and connects to the system $\overline{\text{RESET}}$ from the host processor. When the $\overline{\text{FR}}$ signal is HIGH, the HIGH-to-LOW edge for $\overline{\text{SPR}}$ causes a forced refresh request to occur. Once the $\overline{\text{SPR}}$ signal has gone LOW, all refresh requests made with the $\overline{\text{FR}}$ signal will be disabled. $\overline{\text{SPR}}$ must be kept LOW until $\overline{\text{FR}}$ can be guaranteed to be HIGH. This pin is available only on the surface-mount packages. To implement the function in a DIP or Flatpack package, refer to Figure 4 of the Functional Description.

$\overline{\text{SUP}}$ Suppress (Input; Active LOW)

When $\overline{\text{SUP}}$ is driven LOW, it will inhibit access to memory by disabling $\overline{\text{WE}}$. It can be used to prevent illegal access in memory-access-protected systems.

T_{1-10} Timing Taps (Inputs)

These inputs are positive-edge triggered timing tap outputs from the timing reference generator (PEG or delay line). They provide the necessary timing information for the Am2969 to control the sequence of events in any given cycle. Since T_1 is also sampled by T_2, it must meet appropriate setup and hold times relative to T_2. Definition of the ten timing taps is given in Table 2 of the Functional Description. The numbers T_1–T_{10} have no implication to the order in which they are used with the exceptions of T_3, T_6, T_9, and T_{10}, which must appear in that order.

V_{CC} +5-V Positive Power Supply (Input)

Refer to the connection diagram for placement and number of V_{CC} pins. Refer to the Functional Description for information on proper decoupling.

$\overline{\text{WE}}$ Write Enable (Output; Active LOW)

When $\overline{\text{WE}}$ is LOW, it causes data to be written into memory. $\overline{\text{WE}}$ is inhibited if $\overline{\text{SUP}}$ is LOW, or if a multiple error is encountered during a read-modify-write cycle.

FUNCTIONAL DESCRIPTION

The functional description of the Am2969 is divided into several major sections to best explain the full function of the device. These sections are:

- Am2969 Chip-to-Chip Signals (beginning with a system overview)
- Am2969 Chip Operating Modes
- Am2969 System Operating Modes

System Overview

The AMD approach to memory system design provides the designer with flexibility. AMD's building block approach separates system functions onto separate chips. These functions include Dynamic Memory Control, Memory Timing Control, Error Detection and Correction, and System Timing Generation. In this way, a designer can select the functionality and determine the performance to meet the system requirements.

The Am2969 is the building block which performs memory timing control in DRAM systems which use Error Detection and Correction. Its two main functions are to:

1. Generate all control timing signals for an EDC memory system, and
2. Perform arbitration between memory read/write accesses and refresh accesses.

A DRAM system is shown in Figure 2 (a simplified diagram appears in the Applications section of this data sheet). Using this diagram, the reader can follow the chain of events through the chip set. In general terms, the processor will begin an event sequence by generating a cycle request. This input signal, $\overline{\text{CYCREQ}}$, will cause the Am2969 to generate the RASI signal which in turn acts as the "trigger" to the system timing generator. In AMD's approach to system design, this could be the Am2971 Programmable Event Generator or a delay line of desired resolution. The system timing generator will initiate the timing sequences between chips at the desired system resolution, and therefore performance is user-definable.

The generated timing signals are defined as timing taps. The functional description of each tap is defined in Table 2. These taps, which are fed into the Am2969 and the DMC (Am2968A or Am29368) control the time at which the various control signals are generated by the Am2969. Control signals will vary depending upon which type of cycle has been requested (refresh, read, write, read-modify-write, etc.).

TABLE 1. MODE CONTROL SELECTION

MC_1	MC_0	Mode
0	0	Refresh without Scrubbing
0	1	Refresh with Scrubbing or Memory Initialize
1	0	Read/Write
1	1	Clear Refresh Counter

TABLE 2. TIMING TAP DEFINITIONS

Tap No.	Description/Function
1	Controls when $\overline{\text{DTACK}}$ will go active during Fly-By, Flow-Thru with error, and first Bus-Retry cycles.
2	Identifies when the $\overline{\text{ERR}}$ flag is sampled during a read cycle. This timing tap controls the end of RASI for read, full-word-write, initialize, and read-without-scrubbing cycles.
3	Indicates corrected data at the output of the EDC circuit during a read cycle (with a single-bit error) and partial-word-write cycles. Switches the EDC unit from check to generate mode. Latches data into the EDC in the Flow-Thru mode.
4	Controls when $\overline{\text{DTACK}}$ will go active during read/modify/write cycles. Identifies when $\overline{\text{MERR}}$ is sampled.
5	Asserts $\overline{\text{WE}}$ in full-word-write cycles and latches data into EDC circuit by making LEI LOW. Check bits and data must be valid at this time.
6	Indicates when $\overline{\text{WE}}$ becomes active during read/modify/write cycles. At this time, corrected data and check bits must be available on the memory bus. Also applies to refresh-with-scrubbing cycles.
7	Indicates that valid memory data is available at inputs of EDC unit during all read cycles, partial-word-write cycles, and refresh-with-scrubbing cycles.
8	Identifies when a new memory cycle may begin after a read-without-error, full-word-write, initialize, or refresh-without-scrubbing cycle (includes precharge time). This timing tap is the End of Cycle for read-without-error, full-word-write, and refresh-without-scrubbing cycles.
9	Controls the end of RASI (RASI going HIGH to LOW) for read/modify/write cycles. Also applies to refresh-with-scrubbing cycles.
10	Identifies when a new memory cycle may begin after a read/modify/write cycle has been performed (includes precharge time). Also applies to refresh-with-scrubbing cycles.

System Interface

The following section of the functional description deals with the interface of the Am2969 to the rest of the chip set. Figure 1 provides a grouping of the Am2969 signals by function or chip interface. For additional reference, a detailed chip-to-chip interconnect diagram is shown in Figure 2.

Am2969 to DMC Interface

The Am2969 will control the type of mode that the DMC — either the Am2968A for systems using 256K x 1 or 256K x 4 DRAMs, or the Am29368 for systems using 1M x 1 or 1M x 4 DRAMs — will enter. The mode control signals (MC_n) direct the DMC as to which type of cycle to perform. Mode Control functions are given in Table 1. In addition to the Mode Control signals, the RASI signal from the Am2969 acts as an input to the DMC. It initiates the selected cycle sequence for proper DMC control of the address path and generation of the $\overline{RAS}$ and $\overline{CAS}$ signals to the DRAM array.

Am2969 to System Timing Generator Interface

The Am2969 generates a RASI signal after receiving a $\overline{CYCREQ}$ from the processor. In addition to the RASI signal going to the DMC, it also becomes an input to the system timing generator. The Am2971 PEG or a delay line, after being triggered by this RASI signal, will generate 12 timing taps. These taps, which are defined in Table 2, are fed back into the chip set to initiate the sequence of events. Ten of these taps are fed into the Am2969 at the user-defined time during the cycle operation, and control the output signals of the Am2969 dependent upon the type of cycle which has been started.

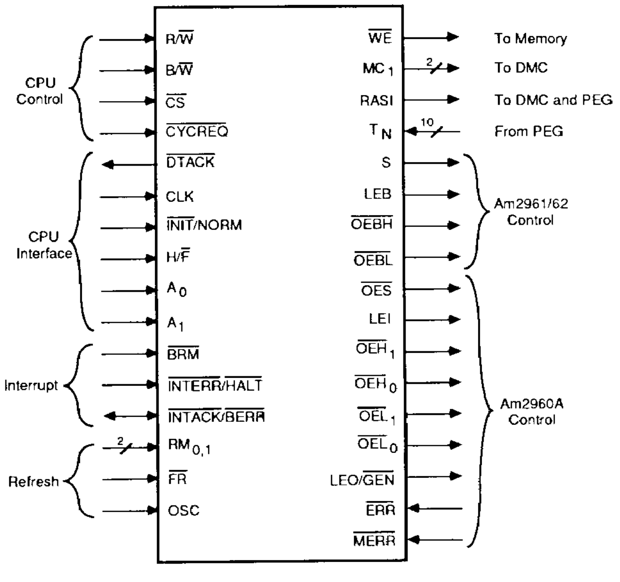
Am2969 to DRAM Array Interface

The Am2969 has only one signal which interfaces directly to the DRAM array. This signal, $\overline{WE}$, is capable of driving up to a 500-pF load and requires no external drivers or damping resistors. $\overline{WE}$ is disabled when $\overline{SUP}$ is used to prevent illegal memory accesses in an access-protected memory system. The $\overline{WE}$ is also inhibited during a read/modify/write cycle if a multiple-error has been detected by the EDC circuit.

Am2969 to Processor Interface

The Am2969 is a general-purpose memory timing controller designed to interface with most processors. Two main handshake signals are used between the processor and the memory system. $\overline{CYCREQ}$ is the request by the processor to the Am2969 to initiate a memory cycle. $\overline{DTACK}$ is the acknowledgment sent to the processor by the Am2969 to indicate that a memory cycle is in progress. Several other signals are used to specify handling of bytes of data. These signals are A_0, A_1, H/$\overline{F}$, and B/$\overline{W}$. Table 3 shows how the enable signals are decoded using these inputs and also shows the decoding for half- and full-word enables using the A_1 and H/$\overline{F}$ signals. Under most conditions, several address signals are used by the processor to generate the chip select signal ($\overline{CS}$). The Read/Write signal (R/$\overline{W}$) from the processor is used to control read and write cycles of operation.

Other miscellaneous processor signals include an Initialize/Normal signal for initialization of the DRAM array, and a Forced Refresh signal for performing refreshes controlled external to the Am2969. All refresh modes can be chosen by hardwiring the three Refresh Mode input lines as indicated in Table 4.



05221C-5

Figure 1. Am2969 Functional Interface

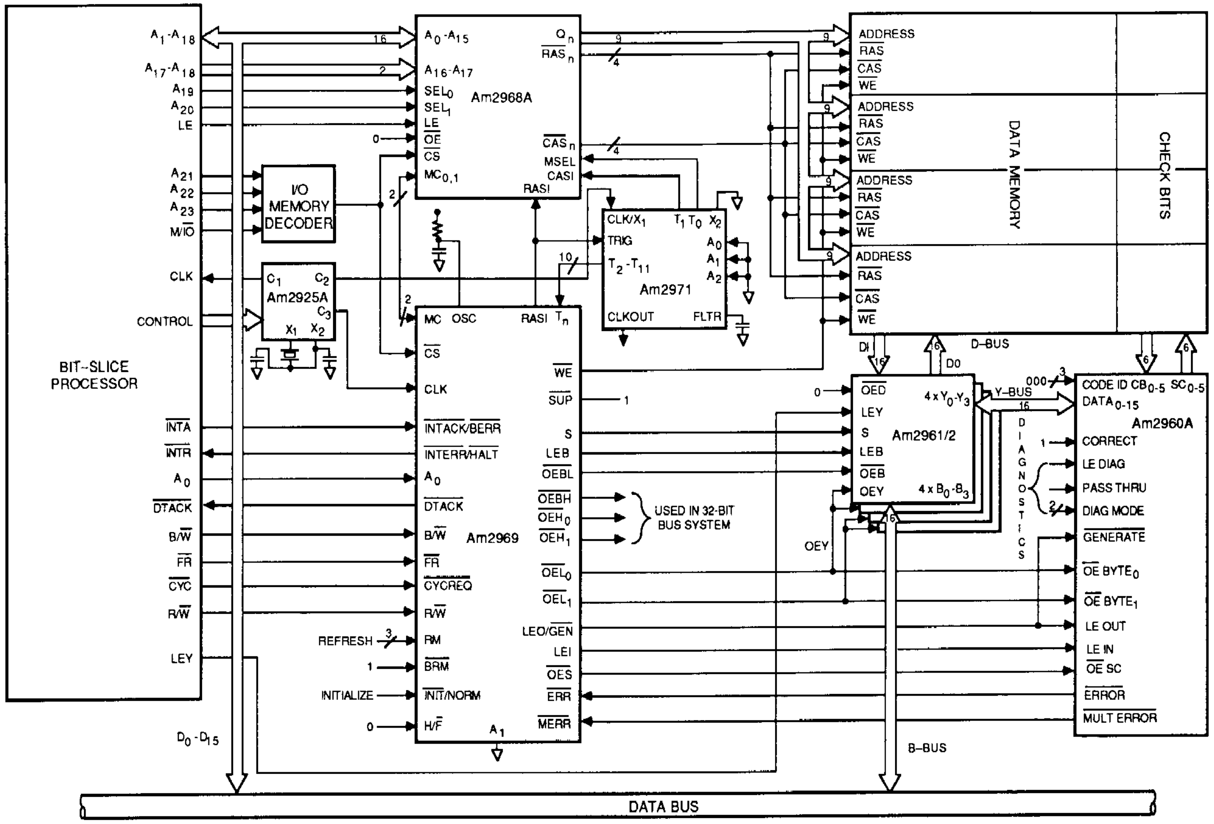


Figure 2. Detailed Chip Hook-Up

05221C-6

The $\overline{\text{INTERR}}$ signal interrupts the CPU when an error occurs, and the $\overline{\text{INTACK}}$ signal from the CPU acknowledges the interrupt.

When interfacing with the Motorola family of microprocessors, or newer generations of microprocessors such as the 80386 or the Am29000, the bus retry feature can be conveniently used with the $\overline{\text{BRM}}$ input hardwired LOW. $\overline{\text{BERR}}$ and $\overline{\text{HALT}}$ signals are asserted simultaneously to indicate an error to the CPU. Under these conditions, the processor will cause the Am2969 to repeat the previous cycle, thus allowing for the correction of the error. This mode minimizes overhead and maximizes performance for CPUs that support this feature.

TABLE 3. OUTPUT ENABLE TRUTH TABLE

H/$\overline{\text{F}}$	B/$\overline{\text{W}}$	A_1	A_0	$\overline{\text{OEH}}_1$	$\overline{\text{OEH}}_0$	$\overline{\text{OEL}}_1$	$\overline{\text{OEL}}_0$	Cycle Type
0	0	0	0	1	1	1	1	Long-Word Write for 32-bit systems or Word Write for 16-bit systems
0	0	0	1	1	1	1	1	
0	0	1	0	1	1	1	1	
0	0	1	1	1	1	1	1	
X	1	0	0	0	0	0	1	Byte Write
X	1	0	1	0	0	1	0	
X	1	1	0	0	1	0	0	
X	1	1	1	1	0	0	0	
1	0	0	X	0	0	1	1	Word Write for 32-bit systems
1	0	1	X	1	1	0	0	
X	X	X	X	0	0	0	0	Read with Error
X	X	X	X	1	1	1	1	Read without Error

Key: X = Don't Care

TABLE 4. REFRESH MODE SELECT TABLE

RM_2	RM_1	RM_0	Refresh Mode
0	0	0	Non-Scrubbing/Non-Burst
0	0	1	Non-Scrubbing/128-Cycle Burst
0	1	0	Non-Scrubbing/256-Cycle Burst
0	1	1	Non-Scrubbing/512-Cycle Burst
1	0	0	Scrubbing/Non-Burst
1	0	1	Scrubbing/128-Cycle Burst
1	1	0	Scrubbing/256-Cycle Burst
1	1	1	Scrubbing/512-Cycle Burst

Am2969 to EDC Interface

The Am2969 also acts as the timing controller for the Am2960 family of EDC circuits and the Am2961/62 EDC Bus Buffers. These direct connections are shown in Figure 2. The $\overline{ERR}$ signal indicates the occurrence of a single-bit error in the DRAM array. If a double-bit (or multiple-bit) error is detected, both the $\overline{ERR}$ and $\overline{MERR}$ signals are activated. The LEI, LEO, and $\overline{OE}$ Byte signals are the data-latch and output enable-control signals. These control signals are generated at user-specified times during the different cycles. The $\overline{GEN}$ signal is used to generate the check-bits in the EDC circuit when in the generate mode. The $\overline{OEL}_0$, $\overline{OEL}_1$, $\overline{OEH}_0$, and $\overline{OEH}_1$ signals are used to enable the different bytes during byte operations. These signals connect to both the EDC circuit and to the EDC Bus Buffers to control the flow of data on the Y-Bus. The Y-Bus connects the EDC unit and the EDC Bus Buffers. When the $\overline{OEL}_n/\overline{OEH}_n$ control signals go LOW, the data flow is from the EDC circuit into the Bus Buffers along the Y-Bus. When HIGH, the data flows along the Y-Bus from the Bus Buffers into the EDC circuit. $\overline{OEL}_n/\overline{OEH}_n$ are connected to the OEY inputs of the Bus Buffers, and $\overline{OE}$, Byte 0, 1 of the EDC circuit. The details of signal decoding are shown in Table 3.

Am2969 to EDC Bus Buffers Interface

The S signal is used to select whether data flow is from the DRAM array or the system bus. The $\overline{OEBL}$ and $\overline{OEBH}$ control the selection of the lower and higher half words of data from the system bus. LEB is the latch-enable signal for data flow between the EDC bus and the system bus. Decoding of $\overline{OEBL}$ and $\overline{OEBH}$ is shown in Table 5.

TABLE 5. OEB TRUTH TABLE

$H/\overline{F}$	A_1	$\overline{OEBH}$	$\overline{OEBL}$
0	X	0	0
1	0	1	0
1	1	0	1

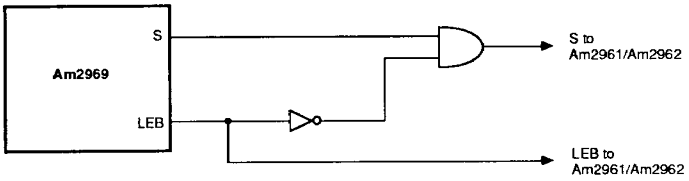
Key: X = Don't Care

When the system has been set up for Read-Without-Error Cycles (Fly-By Mode), the S signal must be modified by external logic when the Am2961/62 Bus Buffers are used. This timing adjustment is required because the Am2969 propagation delay from T_2 (timing tap) to S is faster than the delay from T_2 to LEB. The adjustment is illustrated in Figure 3. Although this will also modify the S signal timing for other cycles, system timing will not be affected.

Am2969 Miscellaneous Signal Interface

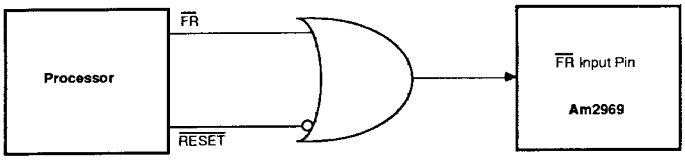
System Power-On Reset (SPR) — The 68-lead surface-mount versions of the Am2969 have an $\overline{SPR}$ input pin which is normally connected to the system or host processor $\overline{RESET}$. Actual initialization of the Am2969 is performed by an internal power-up initialization circuit. The purpose of the $\overline{SPR}$ pin is to inhibit Forced Refreshes from starting cycles before the rest of the system is completely initialized. As long as $\overline{SPR}$ is active, $\overline{FR}$ will be disabled.

When using a 48-pin DIP or 48-pin Flatpack, the $\overline{SPR}$ function must be generated externally as these packages are pin-limited. This simple hook-up is illustrated in Figure 4. In this case, the $\overline{FR}$ pin will be held HIGH to ensure proper functioning of the Am2969 during power-up.



05221C-7

Figure 3. S Signal Interface from Am2969 to Am2961/Am2962 in the Fly-By System Operating Mode



05221C-8

Figure 4. Implementation of $\overline{SPR}$ on a DIP or Flatpack Package

Oscillator Input (OSC) — The Am2969 has an on-chip interval timer for generation of refresh request signals in the Automatic Refresh Mode. The OSC input is connected to a simple resistor-capacitor network as shown in Figure 5. Suggested values are R = 47 kΩ and C = 1000 pF.

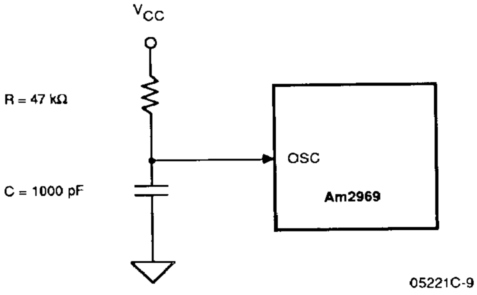


Figure 5. OSC Input R-C Network

Operational Modes

The basic types of operations handled by the Am2969 are read, write, and refresh cycles and arbitration. These operations can be performed in different modes.

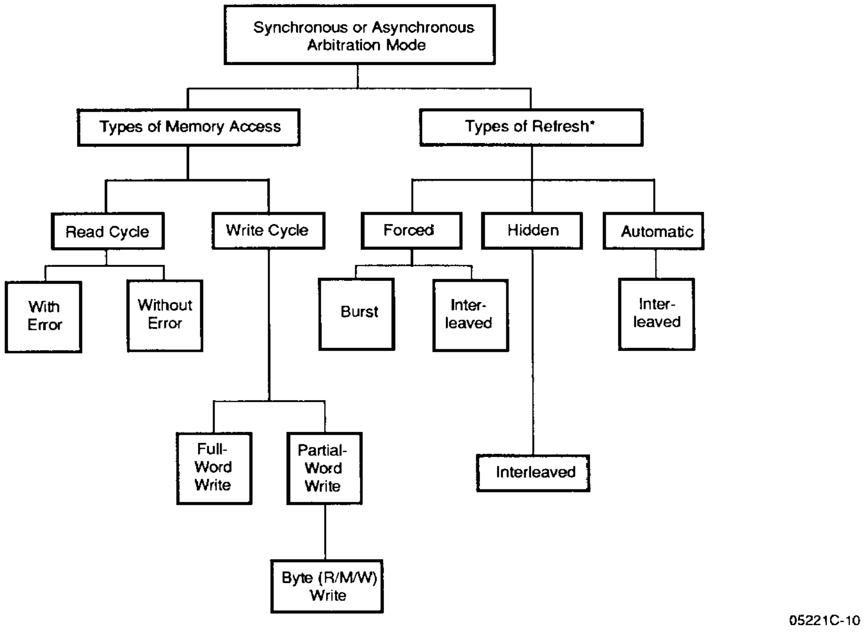
The Am2969 can operate at the chip-level in the following modes:

- Asynchronous Arbitration
- Synchronous Arbitration
- Memory Initialization
- Read and Write Cycles
- Refresh Cycles

Furthermore, the Am2969 can operate in the following system-level modes:

- Flow-Thru
- Fly-By
- Scrubbing
- Bus-Retry

These operating modes are illustrated in the Am2969 Functional Tree of Figure 6. This section of the data sheet will describe the chip-level and system-level modes of the Am2969.



*Refresh cycles can be with or without scrubbing.

Figure 6. Am2969 Functional Operations Tree

Chip-Level Modes of Operation

Synchronous vs Asynchronous Arbitration

A primary function of the Am2969 is to arbitrate between CPU access requests (read/write) and refresh requests. The Am2969 arbitrates memory requests in either synchronous or asynchronous systems.

In either system mode of operation, the processor typically originates the read/write and refresh requests. However, the Am2969, through use of an internal timer, can generate refresh requests independent of the CPU. This frees CPU overhead from data management routines and also provides a "fail-safe" generator of refresh requests to guarantee the integrity of data.

Within the Am2969, arbitration logic decides which request is serviced first when both refresh and access requests are issued simultaneously. The state of the arbiter, in either a synchronous or an asynchronous system, determines when each cycle begins.

Synchronous Arbitration — The Am2969 can perform arbitration in a synchronous system by connection of the system clock signal to the CLK input. A cycle begins with the falling edge of the CLK signal. In the synchronous mode, the Am2969 samples the $\overline{\text{CYCREQ}}$ (read/write) and refresh inputs on the negative of CLK. Should both a refresh and processor-cycle request be issued prior to a negative CLK, both requests will be latched into the arbiter. If both cycle requests occur simultaneously, the arbiter services the processor request first, followed by the refresh request. With the exception of a Burst Refresh, the refresh request is serviced immediately upon completion of the processor cycle. It is not necessary to wait for another CLK edge. A Burst Refresh request is not executed until a falling CLK edge is encountered and there is no processor request. This delay in servicing a Burst Refresh request optimizes processor performance.

Asynchronous Arbitration — The arbiter of the Am2969 also operates in asynchronous systems. Under this condition, every memory request is latched into the arbiter inputs. An internal delay is intended to prevent possible metastable states.

The arbiter switches from synchronous to asynchronous when it detects the absence of transitions on the CLK input. The Am2969 decides that a clock is not present (and therefore operates asynchronously) in either of two ways. First, the asynchronous mode can be initiated by two HIGH-to-LOW transitions of the forced refresh input ($\overline{\text{FR}}$). Alternately, the asynchronous mode can be initiated by issuing three HIGH-to-LOW transitions followed by two LOW-to-HIGH transitions of the oscillator (OSC) input. This second method is accomplished by use of an on-chip 2-bit shift register that is clocked by the internal refresh clock. This register is reset when the CLK is LOW. This implementation allows the clock signal from a synchronous system to be interrupted while the clock is in the logic-HIGH state, and service the refresh request through the asynchronous arbiter.

Either method of initiating the asynchronous mode of arbitration causes the generation of an immediate refresh request. For asynchronous arbitration, the CLK input must be tied HIGH. It is recommended that either of the above-mentioned procedures be performed immediately upon power-up during the initialization of the memory. In the asynchronous mode, the arbiter services a processor request before a refresh request. If a processor cycle is being executed, any refresh request is latched into the refresh logic and executed when the current cycle ends. $\overline{\text{CYCREQ}}$ must be held LOW until the end of the current cycle.

Once the asynchronous mode of arbitration has been initiated, the Am2969 cannot switch to synchronous operation.

Types of Memory Accesses

Memory Initialization

The Am2969 provides the necessary signals to perform memory initialization. Error-correcting memories must be initiated to a known state to avoid read errors associated with memory power-up. The Am2969 facilitates this procedure via the $\overline{\text{INIT}}$/NORM input. The data pattern written into memory can be either all zeros (using the initialization mode of the Am29C60A), or any desired pattern present on the system bus during the first memory cycle of the initialization routine.

Upon entering the Initialize mode, the Am2969 performs eight "wake-up" cycles followed by 2^{20} write/refresh cycles. During this time the processor will remain in a wait state if it attempts to access the memory. Upon completion of the initialization, the Am2969 begins normal memory arbitration.

This initialization is given priority over either processor or refresh requests. The user is strongly urged to take precautions against an accidental initialization request during a normal processor or refresh cycle.

Read and Write Cycles

Following the Functional Tree of the Am2969, it can be observed that the Am2969 can control many types of processor requests (accesses). The two main accesses are read cycles and write cycles. Within the read cycles, the Am2969 can perform read-without-error and read-with-error cycles. Within the write cycles, the Am2969 can perform full-word-write, partial-word-write, and byte-write cycles. All of these cycles are normal read/write cycles except byte-write, which requires an extended read/modify/write cycle.

To initiate a memory access, the processor generates the following signals: $\overline{\text{CS}}$, $\overline{\text{CYCREQ}}$, B/$\overline{\text{W}}$, and R/$\overline{\text{W}}$. The Am2969 subsequently generates the RASI (Row Address Strobe Input) and the MC_1 (Mode Control) signals. Based upon the interface signals, the Am2969 asserts the appropriate $\overline{\text{WE}}$ (Write Enable) and $\overline{\text{DTACK}}$ (Data Transfer Acknowledge) signals at the appropriate times. The Write-Enable output of the Am2969 is designed to directly drive up to 88 DRAMs, 500-pF load. To calculate loads exceeding 500 pF, refer to either the Am2968A or Am29368 data sheets.

Refresh Cycles

The Am2969 can support a variety of refresh methods. The choice of refresh mode is controlled via the RM_n inputs. The basic refresh modes which are supported by the Am2969 are forced, hidden, and automatic refresh. Within these modes there are two methods of execution: interleaved (or distributed) and burst. Burst refreshes can be 128, 256, or 512 cycles in duration.

A forced refresh is an externally requested refresh executable in either the interleaved or burst mode. The refresh signal is generated by external logic which is under the control of the processor. Hidden refresh occurs when the processor is accessing other I/O devices. A hidden refresh can only be performed once every refresh clock period and occurs only as an interleaved refresh. Figure 7 illustrates the system timing for a hidden refresh. Finally, automatic refresh is an Am2969-generated refresh. This "fail-safe" method of refresh is generated every 15.6 μs by an on-chip internal oscillator. The Am2969 effectively controls both internally (automatic) and externally (hidden or forced) refresh requests.

Forced and Automatic Refreshes

An internal refresh request (automatic) is controlled through either the $\overline{\text{FR}}$ input or the internal oscillator.

In the burst mode of operation, the $\overline{\text{FR}}$ input is always the refresh clock.

The $\overline{\text{FR}}$ input can also be used as the refresh clock in the interleaved (distributed) mode. However, the internal oscillator takes over as the refresh clock if it goes through three cycles without a LOW level appearing on the $\overline{\text{FR}}$ input. Therefore, when a forced refresh is not requested within three oscillator clock cycles, an automatic refresh is generated. This allows the primary refresh clock ($\overline{\text{FR}}$) to be interrupted while it is in the HIGH logic state and causes refresh operations to commence at the 15.6 μs internal oscillator rate.

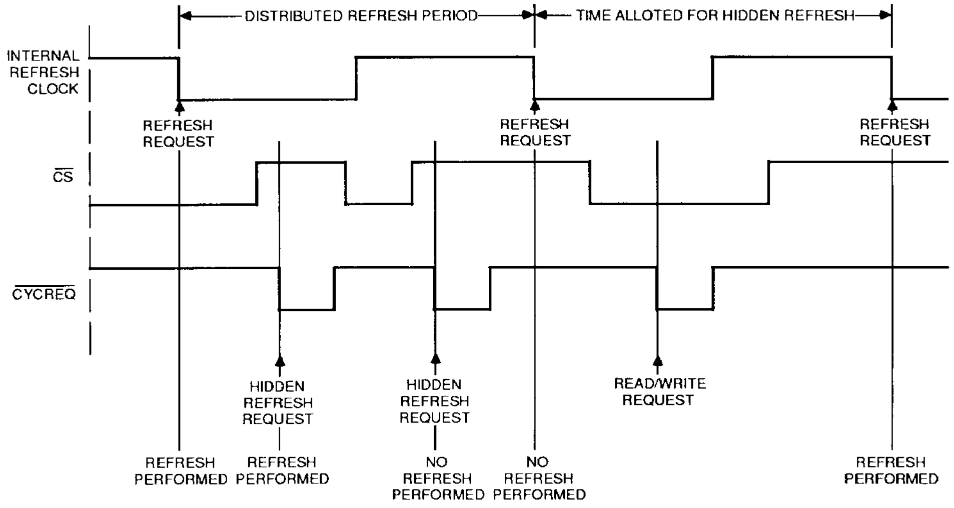
Finally, it is possible to control the refresh requests through the use of the on-chip internal oscillator only when in the interleaved (distributed) mode. In this automatic (or fail-safe) mode, the $\overline{\text{FR}}$ input must be tied HIGH.

Hidden Refresh

The Am2969 can increase memory bandwidth by inserting refresh requests when the processor is accessing other devices or performing I/O operations ($\overline{\text{CYCREQ}}$ = LOW, $\overline{\text{CS}}$ = HIGH). When a hidden refresh is performed, the Am2969 will skip the next refresh request. Only one hidden refresh is allowed for each forced or automatic refresh. Two hidden refresh requests cannot be serviced in succession; they must be interleaved by a forced or automatic refresh request.

A hidden refresh could occur illegally if $\overline{\text{CS}}$ goes LOW-to-HIGH when $\overline{\text{CYCREQ}}$ is LOW. To avoid this condition, assert $\overline{\text{CYCREQ}}$ HIGH before $\overline{\text{CS}}$ goes HIGH. Figure 7 shows the timing involved to perform a hidden-refresh cycle.

Depending upon the system configuration, it is possible for the DRAM to appear "static" to the processor. The Am2969 will take over full control of refresh requests performing a hidden refresh on every clock cycle. Further system optimization (e.g., elimination of wait states) and avoidance of arbitration collisions between refresh and memory requests can be achieved by decoding the I/O status signals and forcing refreshes during I/O operations.



05221C-11

Figure 7. Hidden-Refresh Cycle

System-Level Modes of Operation

Flow-Thru-Read Cycles

The simplest method to implement an error-correcting memory is to extend all read cycles long enough to have single errors detected before the Data Transfer Acknowledge ($\overline{\text{DTACK}}$) is asserted. The Am2960A Error Detection and Correction Circuit assumes every word is in error and completes a full correction routine. It is provided the time to complete this cycle because only read/modify/write cycles are used. When using this Flow-Thru system implementation, faster memory must be used to allow the single error flag to become valid prior to the assertion of $\overline{\text{DTACK}}$. If faster memory is unavailable or undesirable, the Am2969 will insert wait states into the read cycle. This will compromise memory bandwidth, although system timing is simplified.

Fly-By-Read Cycles

Fly-By systems provide the highest data throughput available in an error-correcting system. This is accomplished by assuming that all reads will result in correct data. $\overline{\text{DTACK}}$ is asserted earlier in the cycle and the processor is interrupted only if any error occurs (single-bit or mulitple-bit errors). The EDC circuit does not correct the errors, leaving the CPU to typically enter a recovery routine. Since the occurrence of errors is infrequent, memory bandwidth is improved.

Although Fly-By systems may have an increased number of interrupts sent to the processor, system throughput will be affected minimally as the occurrence of soft errors is generally low. However, a DRAM device failure (hard error) can cause every read to that bank of memory to result in an interrupt, severely affecting system performance. An error-logging function can flag the repeat occurrence and alert the operator of possible hard errors.

Flow-Thru vs Fly-By at the Chip Level

The Am2969 determines whether Flow-Thru or Fly-By has been implemented by the relationship between timing taps T_1 and T_2. The Flow-Thru Mode is identified when T_2 goes HIGH prior to T_1. Similarly, the Fly-By mode is selected if the Am2969 detects timing tap T_1 going HIGH prior to T_2.

The Am2969 can be used in non-EDC systems. However, the Am2970 Memory Timing Controller is designed for such applications. The Am2970 will provide the same control, interval timing, and arbitration as the Am2969, but without the overhead and signals required for the control of the EDC circuit.

Scrubbing

Memory scrubbing is a housekeeping operation in which memory is checked for errors during normal refresh operations. The Am2969 is the only memory timing controller which can facilitate AMD's patented Scrubbing technique.

On each refresh-with-scrub cycle, one memory location is read, checked for errors, and if necessary, corrected before being written back into memory. For a one-megaword memory (2^{20} locations) with one refresh every 16 µs, the Am2969/Am2960A/Am29368 Memory Support chip set will "scrub" the entire memory of single-bit errors every 16 seconds. If multiple-bit errors are encountered during a scrub cycle, $\overline{\text{WE}}$ is suppressed.

With the occurrence of an error, a read/modify/write (R/M/W) cycle is performed. The duration of a R/M/W cycle is typically longer than a normal read or write cycle. During refresh operations, a row in each bank is accessed by energizing the $\overline{\text{RAS}}$ line. This refreshes all the locations in that row. If an error is detected, a write operation is performed in that bank at the location of the error. This is accomplished by energizing the $\overline{\text{CAS}}$ line in that bank for that location. The entire checking operation is performed within the refresh cycle. A wait state may need to be issued to extend the cycle should an error be discovered. However, the system reliability will be increased because soft errors will not be able to accumulate in areas of memory that are not frequently accessed.

When performing refresh without scrubbing, all four $\overline{\text{RAS}}$ lines go LOW but the $\overline{\text{CAS}}$ lines remain HIGH or inactive. A refresh with scrubbing will activate all four $\overline{\text{RAS}}$ lines as before and a single $\overline{\text{CAS}}$ line. Errors that are detected during scrubbing cycles do not cause interrupts or bus-error ($\overline{\text{BERR}}$) signal assertions.

Bus Retry Mode

The Am2969 has a special read access mode for system processors that can support a Bus-Retry request. Bus Retry is an efficient error correction mode usable in both Flow-Thru and Fly-By operations.

When Bus Retry is implemented, read cycles normally are performed as Fly-By. In this mode, the Am2969 does not interrupt the processor when a single-bit error is encountered. Instead, it requests that the last cycle be "re-run" by asserting $\overline{\text{BERR}}$ and $\overline{\text{HALT}}$ simultaneously. An internal-status flip-flop is set that reserves the next processor request for a read cycle to the same location. On this access, $\overline{\text{DTACK}}$ is delayed until the erroneous data is corrected. If a multiple-bit error is detected, $\overline{\text{BERR}}$ is asserted and the processor can enter an error-recovery subroutine.

The relationship of $\overline{\text{ERR}}$, $\overline{\text{MERR}}$, $\overline{\text{HALT}}$, $\overline{\text{BERR}}$, and various timing tap inputs are shown in the Switching Waveforms titled "Bus-Retry Mode, First-Access Timing Diagram" and "Bus-Retry Mode, Second-Access Timing Diagram."

The assertion of $\overline{\text{DTACK}}$ (LOW) can be summarized as follows:

- T_1 causes $\overline{\text{DTACK}}$: Flow-Thru with error cycles, all Fly-By cycles, and first Bus-Retry cycles.
- T_4 causes $\overline{\text{DTACK}}$: Flow-Thru with error cycles.
- RASI causes $\overline{\text{DTACK}}$: Full-word-write cycles.
- T_2 causes $\overline{\text{DTACK}}$: Second Bus-Retry without error cycles.

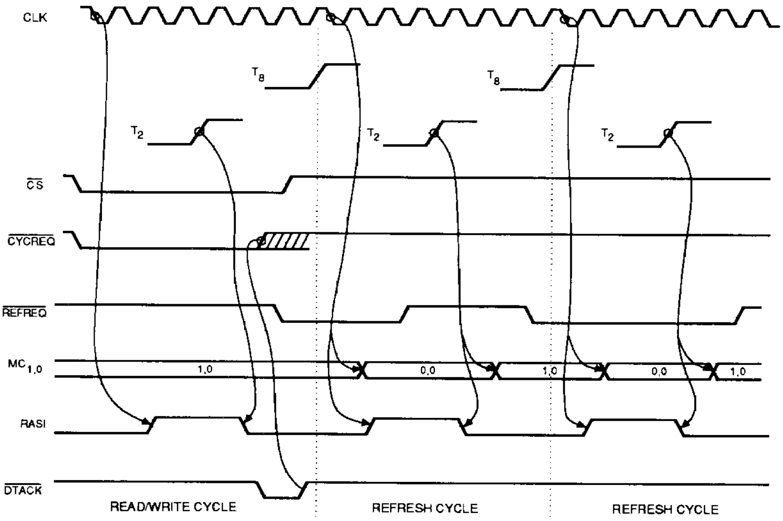
In systems not using Bus Retry, $\overline{\text{INTERR}}$ is generated to interrupt the processor on the detection of a single-bit error in the Fly-By Mode or a multiple-bit error in the Flow-Thru Mode. After an interrupt request is made, $\overline{\text{INTACK}}$ is brought LOW to reset the $\overline{\text{INTERR}}$ output.

Back-to-Back Cycles

Situations may arise which cause repeated memory cycles, one immediately after the other. An example of such a situation is a refresh request occurring during a memory read cycle. Figures 8 and 9 illustrate further examples of Back-to-Back cycles.

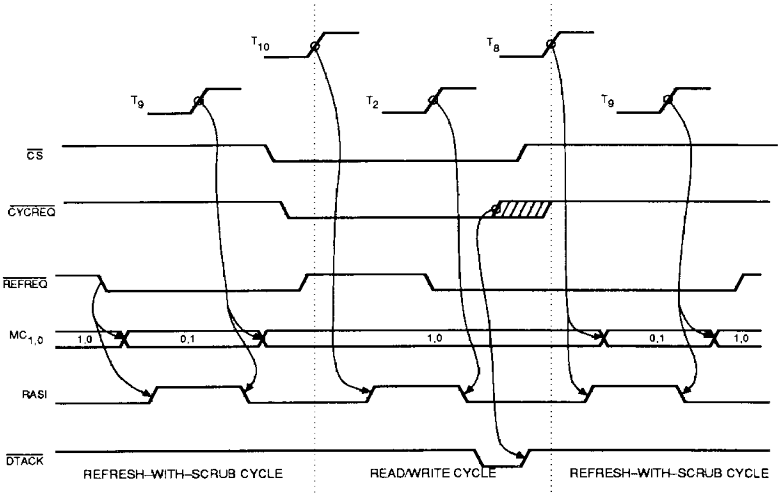
Device Decoupling

Proper decoupling of the Am2969 is necessary for proper operation of the device. The recommended decoupling is illustrated in Figure 10. C_1, C_2, and C_3 should be monolithic capacitors. The capacitors should be mounted as close as possible to the V_{CC} pin(s).



05221C-12

Figure 8. Back-to-Back Synchronous Cycles



05221C-13

Figure 9. Back-to-Back Asynchronous Cycles

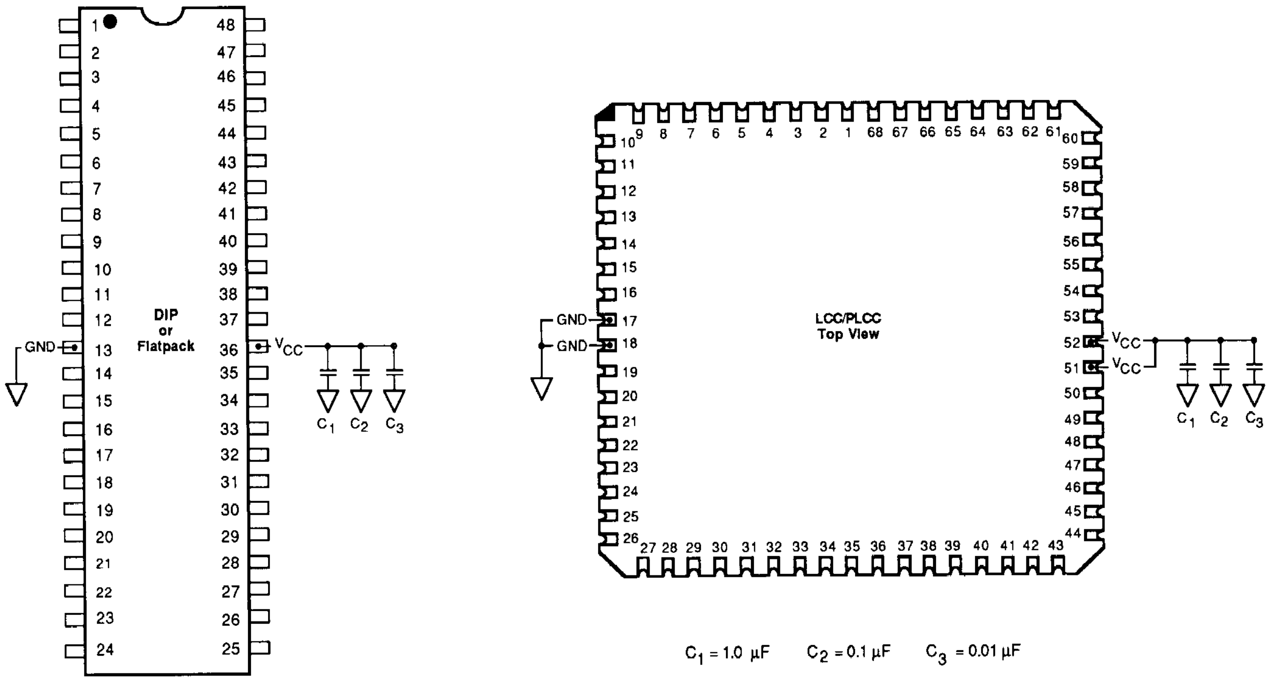


Figure 10. Decoupling Connection Diagrams

05221C-14

APPLICATIONS

Timing Control

To obtain optimum performance and maximum design flexibility, the timing and control logic for the memory system has been kept as a separate function. For systems implementing Error Detection and Correction, the Am2969 Memory Timing Controller (MTC) provides all the necessary control signals for the Am29368, Am2961/62 EDC Bus Buffers, and the Am2960A EDC Unit. The Am2969 Memory Timing Controller uses an Am2972 PEG or delay lines to provide the most accurate timing reference from which the control signals are derived.

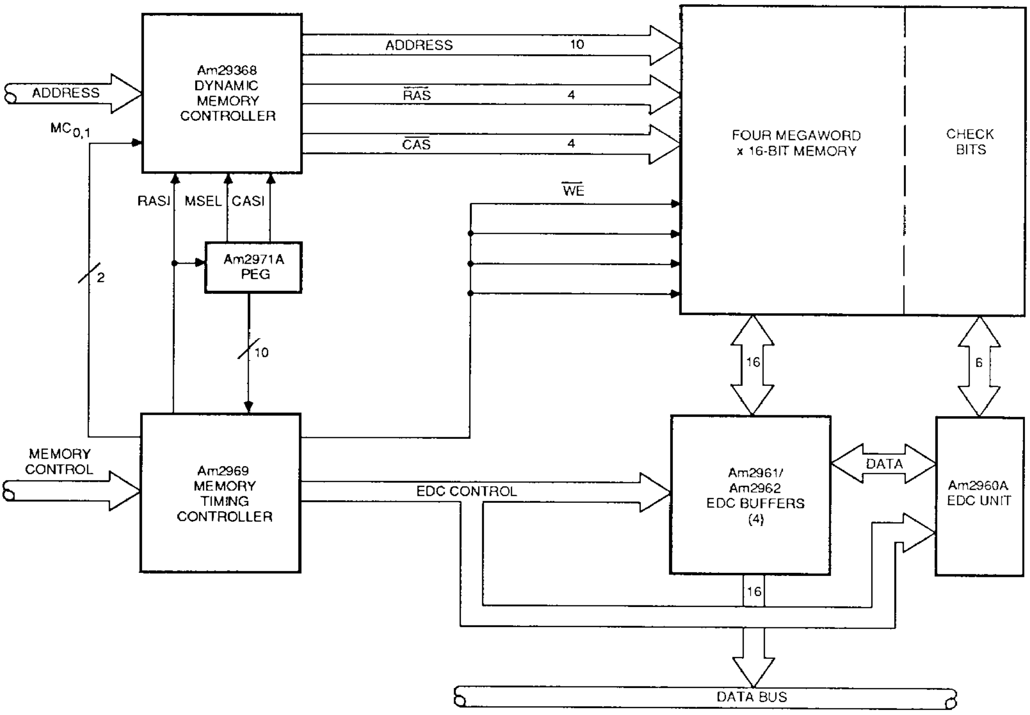


Figure 11. Four-Megaword Dynamic Memory with Error Detection and Correction

ABSOLUTE MAXIMUM RATINGS

Storage Temperature	–65 to +150°C
Ambient Temperature with Power Applied	–55 to +125 C
Supply Voltage to Ground Potential Continuous	–0.5 V to +7.0 V
DC Voltage Applied to Outputs for HIGH Output State	–0.5 V to +V_{CC} Max.
DC Input Voltage	–0.5 V to +5.5 V
DC Input Current	–30 mA to +5.0 mA

Stresses above those listed under ABSOLUTE MAXIMUM RATINGS may cause permanent device failure. Functionality at or above these limits is not implied. Exposure to absolute maximum ratings for extended periods may affect device reliability.

OPERATING RANGES

Commercial (C) Devices	
Ambient Temperature (T_A	0 to +70°C
Supply Voltage (V_{CC})	5.0 V ±10%
Min	4.5 V
Max	5.5 V
Military* (M) and Extended Commercial (E) Devices	
Case Temperature (T_C)	–55 to +125°C
Supply Voltage (V_{CC})	5.0 V ±10%
Min	4.5 V
Max	5.5 V

Operating ranges define those limits between which the functionality of the device is guaranteed.

* Military Product 100% tested at T_C = +25°C, +125°C, and –55 C.

DC CHARACTERISTICS over operating range unless otherwise specified (for APL Products, Group A, Subgroups 1, 2, 3 are tested unless otherwise noted)

Parameter Symbol	Parameter Description	Test Conditions (Note 1)	Min.	Max.	Unit
V_{OH}	Output HIGH Voltage. All Outputs Except: $\overline{\text{INTACK}}/\overline{\text{BERR}}$, $\overline{\text{DTACK}}$, and $\overline{\text{INTERR}}/\overline{\text{HALT}}$	V_{CC} = Min., I_{OH} = –1.0 mA	2.4		V
V_{OL}	Output LOW Voltage. All Outputs Except: $\overline{\text{WE}}$, S, and LEB	V_{CC} = Min., I_{OL} = 8 mA		0.5	V
	Output LOW Voltage. S and LEB Outputs	V_{CC} = Min., I_{OL} = 16 mA		0.5	V
	Output LOW Voltage. $\overline{\text{WE}}$ Output	V_{CC} = Min., I_{OL} = 2.4 mA		0.5	V
		V_{CC} = Min., I_{OL} = 12 mA		0.8	V
V_{IH}	Input HIGH Level.		2.0		V
V_{IL}	Input LOW Level.			0.8	V
V_I	Input Clamp Voltage.	V_{CC} = Min., I_{IN} = –18 mA		–1.2	V
I_{IL}	Input LOW Current. Inputs: $\overline{\text{ERR}}$, $\overline{\text{MERR}}$, T_4, T_1, T_2, $\overline{\text{CS}}$, RM_2, $\overline{\text{SUP}}$, $\overline{\text{CYCREQ}}$, $\overline{\text{BRM}}$, A_0, A_1, $\text{B}/\overline{\text{W}}$, $\text{R}/\overline{\text{W}}$, $\text{H}/\overline{\text{F}}$, and $\overline{\text{SPR}}$ (Note 2)	V_{CC} = Max., V_{IN} = 0.4 V		–800	µA
	Input LOW Current. Inputs: $\overline{\text{FR}}$, $\overline{\text{INTACK}}$, $\overline{\text{INIT}}$/NORM, CLK, RM_0, RM_1, T_3, and T_5–T_{10}	V_{CC} = Max., V_{IN} = 0.4 V		–400	µA
I_{IH}	Input HIGH Current. Inputs: $\overline{\text{ERR}}$, $\overline{\text{MERR}}$, T_4, $\underline{T_1}$, T_2, $\overline{\text{CS}}$, RM_2, $\overline{\text{SUP}}$, $\overline{\text{CYCREQ}}$, $\overline{\text{BRM}}$, A_0, A_1, $\text{B}/\overline{\text{W}}$, $\text{R}/\overline{\text{W}}$, and $\text{H}/\overline{\text{F}}$.	V_{CC} = Max., V_{IN} = 2.7 V		40	µA
	Input HIGH Current. Inputs: $\overline{\text{FR}}$, $\overline{\text{INTACK}}$, $\overline{\text{INIT}}$/NORM, CLK, RM_0, RM_1, T_3, and T_5–T_{10}.	V_{CC} = Max., V_{IN} = 2.7 V		20	µA
	Input HIGH Current. Input: $\overline{\text{SPR}}$ (Note 2)	V_{CC} = Max., V_{IN} = 2.7 V		–250	µA

DC CHARACTERISTICS (Cont'd.)

Parameter Symbol	Parameter Description	Test Conditions (Note 1)	Min.	Max.	Unit
I_I	Input HIGH Current.	V_{CC} = Max., V_{IN} = 5.5 V		1.0	mA
I_{SC}	Output Short-Circuit Current. All Outputs Except: $\overline{\text{INTACK}}/\overline{\text{BERR}}$, $\overline{\text{DTACK}}$, and $\overline{\text{INTERR}}/\overline{\text{HALT}}$ (Note 3)	V_{CC} = Max., V_{OUT} = 0.0 V	–60	–200	mA
I_{OX}	Open-Collector Output Current. Outputs: $\overline{\text{INTERR}}/\overline{\text{BERR}}$, $\overline{\text{DTACK}}$, and $\overline{\text{INTACK}}/\overline{\text{HALT}}$	V_{CC} = Min., V_{OH} = 5.5 V		250	µA
I_{CC}	Power Supply Current. (Note 4)	V_{CC} = 5.0 V, T_A = 25°C		185	mA
		V_{CC} = Max.		260	

Notes: 1. For conditions shown as Min. or Max., use the appropriate value specified under Operating Range for the applicable device type. No input tests are performed on the OSC input.
2. $\overline{\text{SPR}}$ is available only on surface-mount packages.
3. Not more than one output should be shorted together at a time. Duration of the short-circuit test should not exceed one second.
4. Typical I_{CC} (V_{CC} = 5.0 V, T_A = 25°C) is neither tested nor guaranteed, but should be representative of a nominal unit.

SWITCHING CHARACTERISTICS over operating range unless otherwise specified (for APL Products, Group A, Subgroups 9, 10, 11 are tested unless otherwise noted) (C_L = 50 pF)

No.	Parameter Symbol	Parameter Description	System Condition		Min.	Max.	Unit
1	t_{PD}	CLK ↓ to RASI ↑	Synchronous R/W Cycles (see Diagram A)			24	ns
2	t_{PD}	$\overline{\text{CYCREQ}}$ ↓ to RASI ↑	Asynchronous R/W Cycles (see Diagram B)			28	ns
3	t_{PD}	T_8 ↑ to RASI ↑	Asynchronous Back-to-Back R/W and Initialize Cycles (see Diagram M)			57	ns
4	t_{PD}	T_{10} ↑ to RASI ↑	Asynchronous Back-to-Back R/W Cycles			55	ns
5	t_S	$MC_{0,1}$ (Note 1) to RASI ↑	All Refresh Cycles (see Diagram I)	COM'L	18	55	ns
				MIL and E-COM'L	18	58	
6	t_H	$MC_{0,1}$ (Note 1) to RASI ↓	All Refresh Cycles (see Diagram I)		5		ns
7	t_{PD}	T_2 ↑ to RASI ↓	All Short Cycles (see Diagram B)			25	ns
8	t_{PD}	T_9 ↑ to RASI ↓	All Long Cycles (see Diagram C)			20	ns
9	t_{PD}	RASI ↑ to S ↓	All Cycles Except Full-Word-Write and Initialize (see Diagram B)			10	ns
10	t_{PD}	T_2 ↑ to S ↑	All Cycles Except Full-Word-Write and Initialize (see Diagram B)			18	ns

SWITCHING CHARACTERISTICS (Cont'd.)

No.	Parameter Symbol	Parameter Description	System Condition			Min.	Max.	Unit
11	t_{PD}	T_2 ↑ to $\overline{OEH}_n$, $\overline{OEL}_n$ ↓	All R/M/W Cycels (see Diagram C)				33	ns
12	t_{PD}	T_8 ↑ to $\overline{OEH}_n$, $\overline{OEL}_n$ ↓	Initialize Cycles (see Diagram M)				40	ns
13	t_{PD}	T_8 ↑ to $\overline{OEH}_n$, $\overline{OEL}_n$	Initialize Cycles (see Diagram N)				37	ns
14	t_{PD}	T_{10} ↑ to $\overline{OEH}_n$, $\overline{OEL}_n$ ↑	All R/M/W Cycles (see DiagramC)				37	ns
15	t_{PD}	RASI ↑ to LEI ↑	R/M/W, Full-Word-Write, and Initialize Cycles (see Diagram C)				12	ns
16	t_{PD}	LEO/$\overline{GEN}$ (Note 1) ↓ to LEI ↑	All R/M/W Cycles (see Diagram C)			5		ns
17	t_{PD}	T_5 ↑ to LEI ↓	Full-Word-Write Cycle (see Diagram H)				20	ns
18	t_{PD}	T_7 ↑ to LEI ↓	All R/M/W Cycles (see Diagram B)				20	ns
19	t_{PD}	T_9 ↑ to LEI ↓	All R/M/W Cycles (see Diagram C)				37	ns
20	t_{PD}	RASI ↑ to LEB ↑	(see Diagram B)				12	ns
21	t_{PD}	T_3 ↑ to LEB ↓	Flow-Thru-Read Cycles (see Diagram C)				21	ns
22	t_{PD}	T_2 ↑ to LEB ↓	Fly-By-Read Cycles (see Diagram B)				30	ns
23	t_{PD}	T_7 ↑ to $\overline{OEBH}$, $\overline{OEBL}$ ↓	(see Diagram B)				16	ns
24	t_{PD}	$\overline{CYCREQ}$ ↑ to $\overline{OEBH}$, $\overline{OEBL}$ ↑					27	ns
25	t_{PD}	RASI ↑ to LEO/$\overline{GEN}$ ↓	Full-Word-Write Cycle (see Diagram H)				12	ns
26	t_{PD}	T_8 ↑ to LEO/$\overline{GEN}$ ↓	Initialize Cycle (see Diagram M)				34	ns
27	t_{PD}	T_3 ↑ to LEO/$\overline{GEN}$ ↓	All R/M/W Cycles (see Diagram C)				18	ns
28	t_{PD}	T_2 ↑ to LEO/$\overline{GEN}$ ↑	Full-Word-Write Cycle (see Diagram H)				33	ns
29	t_{PD}	T_8 ↑ to LEO/$\overline{GEN}$ ↑	Initialize Cycle (see Diagram N)				33	ns
30	t_{PD}	T_9 ↑ to LEO/$\overline{GEN}$ ↑	All R/M/W Cycles (see Diagram C)				33	ns
31	t_{PD}	RASI ↑ to $\overline{WE}$ ↓	Initialize Cycle (see Diagram M)	C_L = 50 pF			6	ns
				C_L = 150 pF			11	ns
				C_L = 500 pF	COM'L		25	ns
					MIL and E-COM'L		27	ns
32	t_{PD}	T_5 ↑ to $\overline{WE}$ ↓	Full-Word-Write Cycle (see Diagram H)	C_L = 50 pF			15	ns
				C_L = 150 pF			22	ns
				C_L = 500 pF	COM'L		36	ns
					MIL and E-COM'L		39	ns

SWITCHING CHARACTERISTICS (Cont'd.)

No.	Parameter Symbol	Parameter Description	System Condition			Min.	Max.	Unit
33	t_{PD}	T_6 ↑ to $\overline{\text{WE}}$ ↓	All R/M/W Cycles (see Diagram C)	C_L = 50 pF			15	ns
				C_L = 150 pF			22	ns
				C_L = 500 pF	COM'L		36	ns
					MIL and E-COM'L		39	ns
34	t_{PD}	T_2 ↑ to $\overline{\text{WE}}$ ↑	Full-Word-Write Cycle (see Diagram H)	C_L = 50 pF			35	ns
				C_L = 150 pF			40	ns
				C_L = 500 pF	COM'L		60	ns
					MIL and E-COM'L		68	ns
35	t_{PD}	T_8 ↑ $\overline{\text{WE}}$ ↑	Initialize Cycle (see Diagram N)	C_L = 50 pF			35	ns
				C_L = 150 pF			40	ns
				C_L = 500 pF	COM'L		60	ns
					MIL and E-COM'L		68	ns
36	t_{PD}	T_9 ↑ to $\overline{\text{WE}}$ ↑	All R/M/W Cycles (see Diagram C)	C_L = 50 pF			35	ns
				C_L = 150 pF			40	ns
				C_L = 500 pF	COM'L		60	ns
					MIL and E-COM'L		68	ns
37	t_{PD}	RASI ↑ to $\overline{\text{OES}}$ ↓	Full-Word-Write and Initialize Cycles (see Diagram H)				12	ns
38	t_{PD}	T_7 ↑ to $\overline{\text{OES}}$ ↓	(see Diagram B)				38	ns
39	t_{PD}	T_8 ↑ to $\overline{\text{OES}}$ ↑	(see Diagram B)				33	ns
40	t_{PD}	T_{10} ↑ to $\overline{\text{OES}}$ ↑	(see Diagram C)				33	ns
41	t_{PD}	T_1 ↑ to $\overline{\text{DTACK}}$ ↓	Fly-By, Flow-Thru-without-Error, and First-Bus-Retry Modes (see Diagram B)				15	ns
42	t_{PD}	T_2 ↑ to $\overline{\text{DTACK}}$ ↓	Second-Bus-Retry Cycle without Error (see Diagram E)				15	ns
43	t_{PD}	T_4 ↑ to $\overline{\text{DTACK}}$ ↓	Flow-Thru with Error Cycles (see Diagram C)				16	ns
44	t_{PD}	RASI ↑ to $\overline{\text{DTACK}}$ ↓	Full-Word-Write Cycle (see Diagram H)				17	ns
45	t_{PD}	$\overline{\text{CYCREQ}}$ ↑ to $\overline{\text{DTACK}}$ ↑	(see Diagram B)				39	ns

SWITCHING CHARACTERISTICS (Cont'd.)

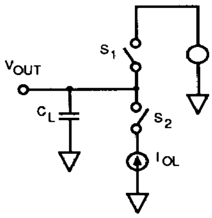
No.	Parameter Symbol	Parameter Description	System Condition	Min.	Max.	Unit
46	t_{PD}	T_2 ↑ to $\overline{BERR}$ ↓	First-Bus Access, $\overline{ERR}$ = LOW (see Diagram D)		16	ns
47	t_{PD}	T_4 ↑ to $\overline{BERR}$ ↓	Second-Bus Access, $\overline{MERR}$ = LOW (see Diagram E)		16	ns
48	t_{PD}	$\overline{CYCREQ}$ ↑ to $\overline{BERR}$↑	(see Diagram D)		40	ns
49	t_{PD}	T_2 ↑ to $\overline{HALT}$ ↓	First-Bus Access, $\overline{ERR}$ = LOW (see Diagram D)		16	ns
50	t_{PD}	$\overline{CYCREQ}$ ↑ to $\overline{HALT}$ ↑	(see Diagram D)		37	ns
51	t_{PD}	T_2 ↑ to $\overline{INTERR}$ ↓	Fly-By Mode, $\overline{ERR}$ = LOW (see Diagram F)		16	ns
52	t_{PD}	T_4 ↑ to $\overline{INTERR}$ ↓	Flow-Thru Mode, $\overline{MERR}$ = LOW (see Diagram G)		16	ns
53	t_{PD}	$\overline{INTACK}$ ↓ to $\overline{INTERR}$ ↑	(see Diagram F)		40	ns
54	t_{PD}	$\overline{CYCREQ}$ ↓ to $MC_{0,1}$	Hidden-Refresh Cycles (see Diagram K)		46	ns
55	t_{PD}	CLK ↓ to $MC_{0,1}$	Refresh Cycles Only (see Diagram L)		24	ns
56	t_{PD}	$\overline{FR}$ ↓ to $MC_{0,1}$	Refresh Cycles Only (see Diagram I)		42	ns
57	t_{PD}	T_8 ↑ to $MC_{0,1}$	Back-to-Back Refresh Cycles (see Diagram B)		50	ns
58	t_{PD}	T_{10} ↑ (Note 1) to $MC_{0,1}$	Back-to-Back Refresh Cycles (see Diagram C)		50	ns
59	t_S	$\overline{CS}$ ↓ to $\overline{CYCREQ}$ ↓	Read/Write Cycles (see Diagram A)	0		ns
60	t_S	$\overline{CS}$ ↑ to $\overline{CYCREQ}$ ↓	Hidden-Refresh Cycles (see Diagram K)	0		ns
61	t_H	$\overline{CS}$ ↑ to $\overline{CYCREQ}$ ↑	Read/Write Cycles (see (Diagram B)	5		ns
62	t_H	$\overline{CS}$ ↓ to $\overline{CYCREQ}$ ↓	Hidden-Refresh Cycles Only (see Diagram K)	23		ns
63	t_S	$A_{0,1}$ or B/$\overline{W}$ to $\overline{CYCREQ}$ ↓	(see Diagram B)	0		ns
64	t_H	$A_{0,1}$ or B/$\overline{W}$ to RASI ↑	(see Diagram B)	4		ns
65	t_S	$\overline{CYCREQ}$ ↓ to CLK ↓	Read/Write Cycles (see Diagram A)	15		ns
66	t_S	$\overline{CYCREQ}$ ↓ to CLK ↓	Hidden-Refresh Cycles (see Diagram L)	27		ns
67	t_S	$\overline{FR}$ ↓ to CLK ↓	Refresh Cycles (see Diagram A)	33		ns
68	t_S	T_8 ↑ to CLK ↓	Back-to-Back Cycles	30		ns
69	t_S	T_{10} ↑ to CLK ↓	Back-to-Back Cycles	30		ns
70	t_H	$\overline{CS}$ ↓ to CLK ↓	Hidden-Refresh Cycles (see Diagram L)	10		ns

SWITCHING CHARACTERISTICS (Cont'd.)

No.	Parameter Symbol	Parameter Description	System Condition	Min.	Max.	Unit
71	t_S	T_2 ↑ to T_1 ↑	Flow-Thru Cycles (see Diagram G)	5		ns
72	t_S	T_1 ↑ to T_2 ↑	Fly-By Cycles (see Diagram D)	5		ns
73	t_H	T_1 ↑ to T_2 ↑	Fly-By Cycles (see Diagram D)	1		ns
74	t_S	$\overline{\text{ERR}}$ to T_2 ↑	(see Diagram B)	12		ns
75	t_H	$\overline{\text{ERR}}$ to T_2 ↑	(see Diagram B)	14		ns
76	t_S	$\overline{\text{MERR}}$ to T_4 ↑	(see Diagram C)	11		ns
77	t_H	$\overline{\text{MERR}}$ to T_4 ↑	(see Diagram C)	13		ns
78	t_S	R/$\overline{\text{W}}$ or H/$\overline{\text{F}}$ (Note 2) to $\overline{\text{CYCREQ}}$ ↓	Asynchronous R/W Cycle (see Diagram B)	0		ns
79	t_S	R/$\overline{\text{W}}$ or H/$\overline{\text{F}}$ (Note 2) to CLK ↓	Synchronous R/W Cycle (see Diagram A)	0		ns
80	t_H	R/$\overline{\text{W}}$ or H/$\overline{\text{F}}$ to T_2 or $\overline{\text{DTACK}}$	(see Diagram A)	21		ns
81	t_S	$\overline{\text{SUP}}$ ↓ to T_5 ↑	Full-Word-Write Cycle (see Diagram H)	5		ns
82	t_S	$\overline{\text{SUP}}$ ↓ to T_6 ↑	R/M/W Cycles (see Diagram C)	5		ns
83	t_H	$\overline{\text{SUP}}$ ↑ to T_2 ↑	Full-Word-Write Cycle (see Diagram H)	28		ns
84	t_H	$\overline{\text{SUP}}$ ↑ to T_9 ↑	R/M/W Cycles (see Diagram C)	25		ns
85	t_S	T_3 ↑ to T_6 ↑	R/M/W and Refresh-with-Scrub Cycle (see Diagram J)	15		ns
86	t_S	T_6 ↑ to T_9 ↑	R/M/W and Refresh-with-Scrub Cycle (see Diagram J)	15		ns
87	t_S	T_9 ↑ to T_{10} ↑	R/M/W and Refresh-with-Scrub Cycle (see Diagram J)	15		ns
88	t_{PW}	T_{2-10} or $\overline{\text{INIT}}$ Pulse Width	(see Diagrams D and E)	10		ns
89	t_{PW}	CLK Pulse Width	CLK = HIGH or LOW (see Diagram A)	10		ns
90	t_{PW}	$\overline{\text{CYCREQ}}$ Pulse Width	Hidden-Refresh Cycles (see Diagram K)	10		ns
91	t_{PW}	$\overline{\text{FR}}$ Pulse Width	Forced-Refresh Cycles (see Diagram A)	10		ns

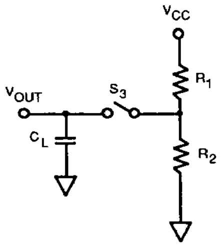
Notes: 1. Parameter is not tested, but is guaranteed by characterization and correlation.
2. Parameter is related to and dependent upon system conditions and therefore cannot be tested. The limit must be met through external system timing to guarantee proper AC operation of the Am2969.

SWITCHING TEST CIRCUITS



05221C-16

Test Circuit A



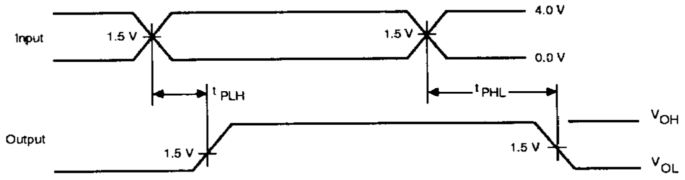
Test Circuit B

Test Output Loads						
Outputs	Test Circuit	R_1 (Ohm)	R_2 (Ohm)	C_L (pF)	I_{OH} (mA)	I_{OL} (mA)
MC$_1$, MC$_0$, RASI, $\overline{\text{OES}}$, LEO/$\overline{\text{GEN}}$, LEI, $\overline{\text{OEBL}}$, $\overline{\text{OEBH}}$, $\overline{\text{OEL}}_1$, $\overline{\text{OEL}}_0$, $\overline{\text{OEH}}_1$, $\overline{\text{OEH}}_0$	A			50	–1	8
S, LEB	A			50	–1	16
$\overline{\text{WE}}$	A			50, 150, 500	–1	12
$\overline{\text{INTERR/HALT}}$, $\overline{\text{INTACK/BERR}}$, $\overline{\text{DTACK}}$	B	680	1600	50		

Notes: 1. C_L, the load capacitance, includes scope probe, wiring, and stray capacitance without the device in the test fixture.

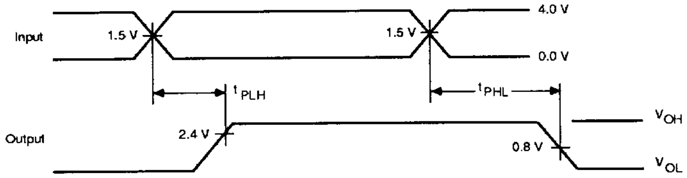
2. S_1, S_2, and S_3 are open during all DC testing.
3. During AC Function testing, switches are set as follows:
   a) S_3 is closed
   b) For $V_{OUT} > 1.5$ V, S_1 is closed and S_2 is open
   c) For $V_{OUT} < 1.5$ V, S_1 is open and S_2 is closed
4. $\overline{\text{INTERR/HALT}}$, $\overline{\text{INTACK/BERR}}$, and $\overline{\text{DTACK}}$ Load:
   a) $V_{CC} = 4.5, 5.0$, and 5.5 V
   b) R_1 is selected to give I_{OL} (Max.) with $V_{CC} = 5.5$ V
   c) R_2 is selected to give V_{OUT} (DC) = 0.7 V_{CC} when the output $\overline{\text{INTERR/HALT}}$, $\overline{\text{INTACK/BERR}}$, and $\overline{\text{DTACK}}$ are tested.

SWITCHING TEST WAVEFORMS



05221C-17

All Outputs Except $\overline{WE}$

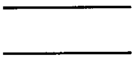
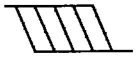
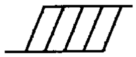
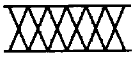


05221C-18

$\overline{WE}$ Output

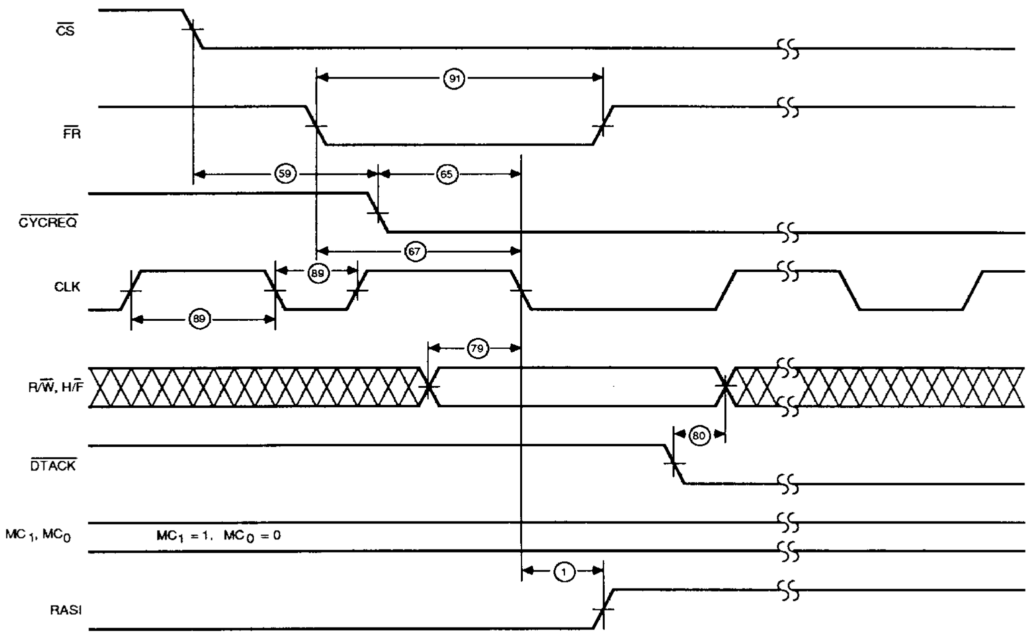
SWITCHING WAVEFORMS

KEY TO SWITCHING WAVEFORMS

WAVEFORM	INPUTS	OUTPUTS
	MUST BE STEADY	WILL BE STEADY
	MAY CHANGE FROM H TO L	WILL BE CHANGING FROM H TO L
	MAY CHANGE FROM L TO H	WILL BE CHANGING FROM L TO H
	DON'T CARE, ANY CHANGE PERMITTED	CHANGING, STATE UNKNOWN

KS000010

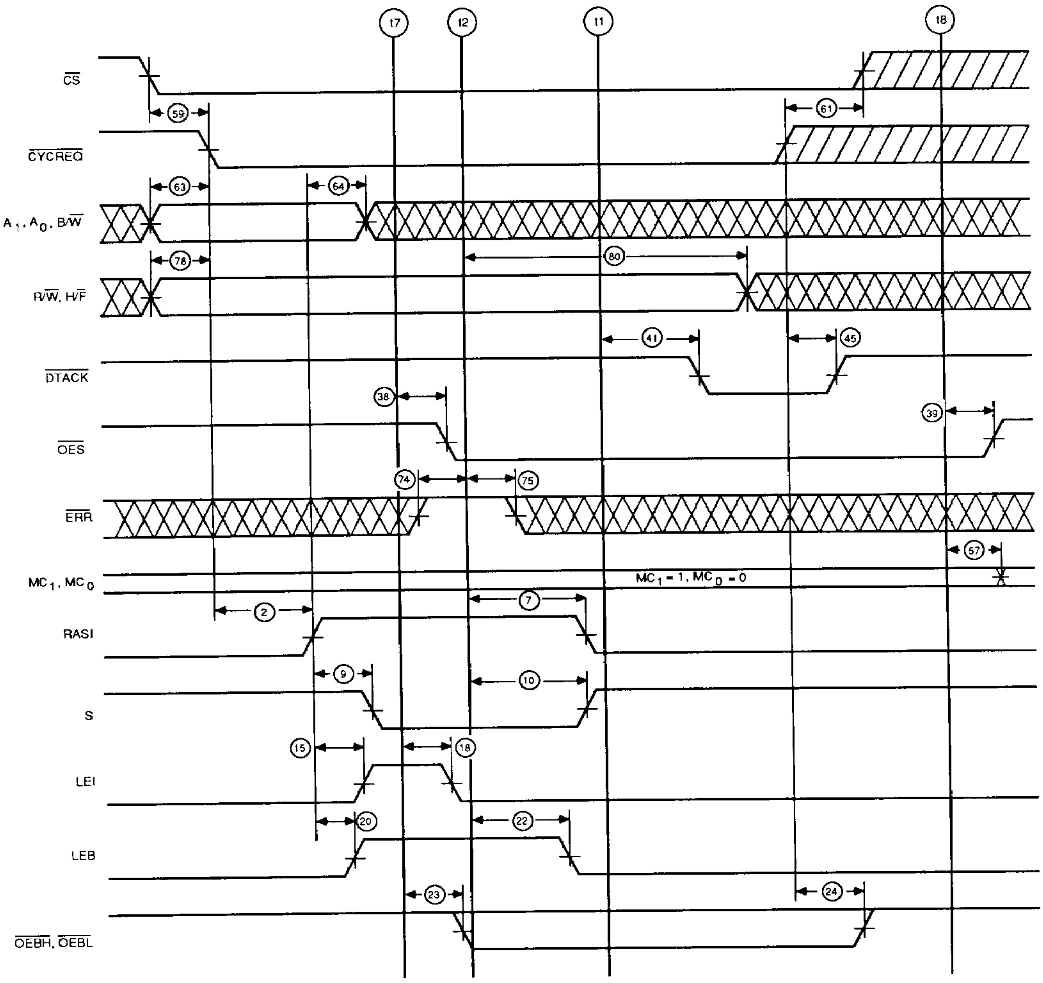
SWITCHING WAVEFORMS (Cont'd.)



05221C-19

Diagram A. Synchronous Arbitration

SWITCHING WAVEFORMS (Cont'd.)

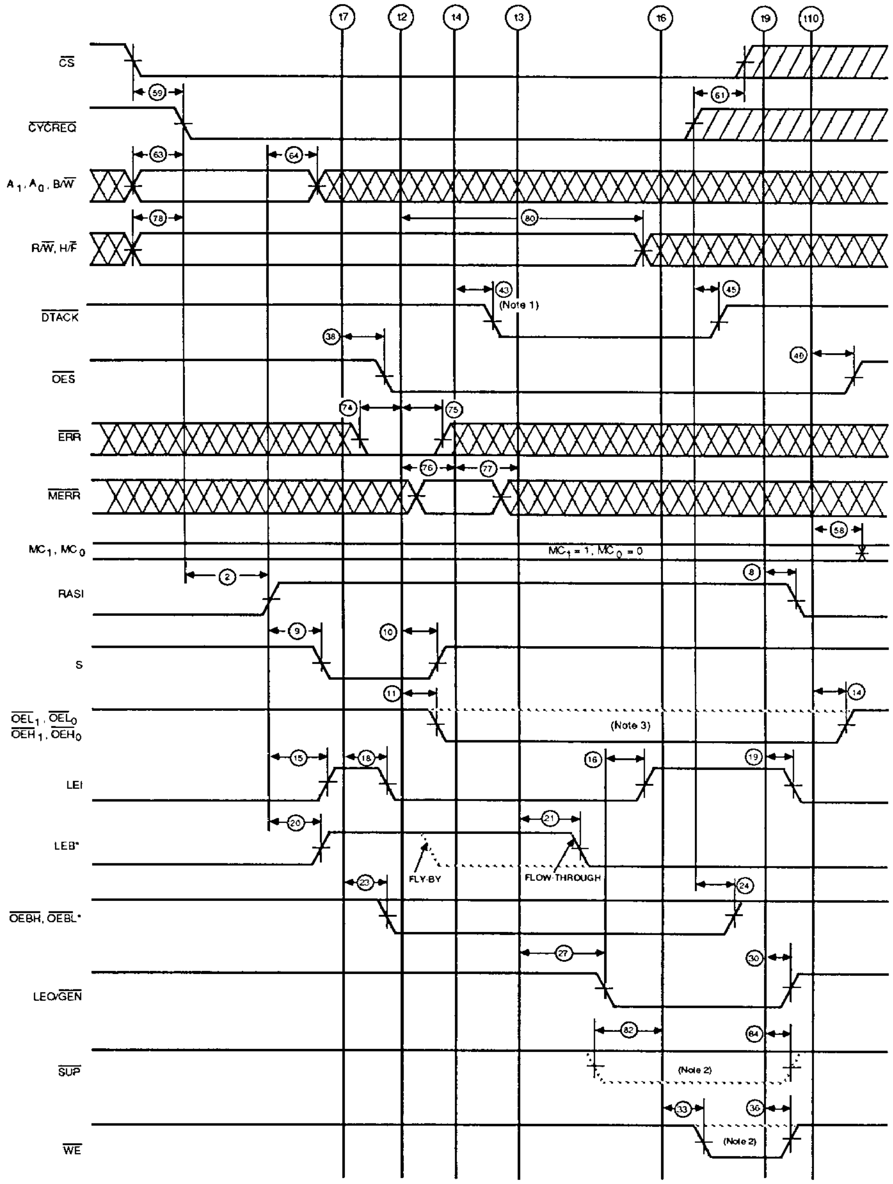


$OEL_n = 1$
$OEH_n = 1$
$LEO/\overline{GEN} = 1$

05221C-20

Diagram B. Read-Without-Error Cycle Timing

SWITCHING WAVEFORMS (Cont'd.)



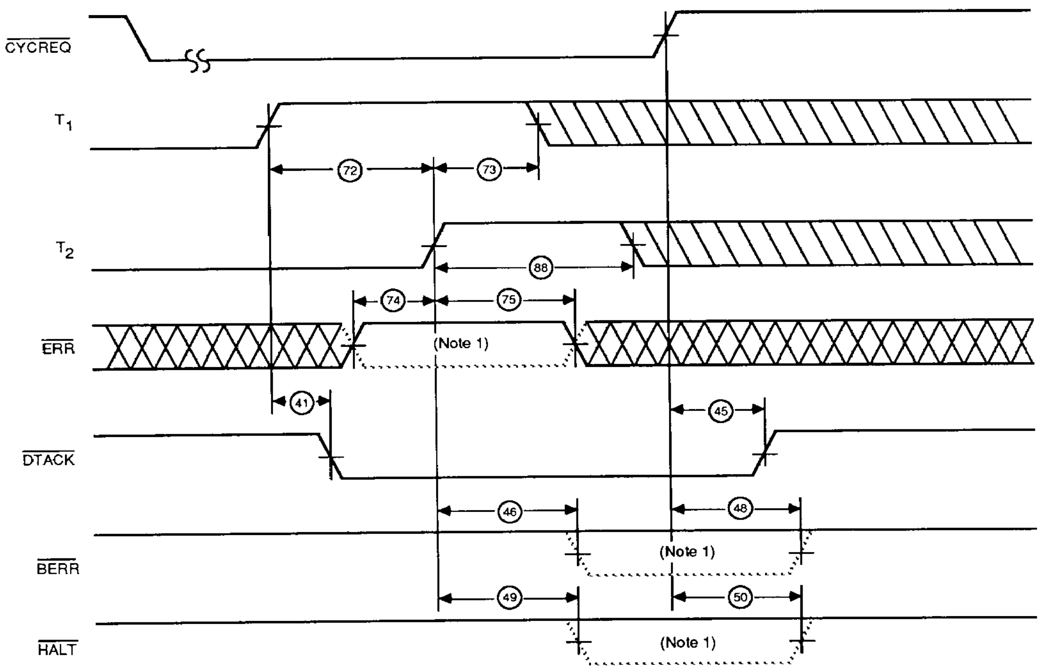
05221C-21

Notes: 1. $\overline{\text{DTACK}}$ is shown for Flow-Through mode. See Figure 5a for Fly-By mode.
2. $\overline{\text{WE}}$ will remain HIGH if SUP is asserted.
3. For Partial-Word Write, these outputs may be HIGH depending on $B/\overline{W}$, $H/\overline{F}$, A_0 and A_1. For Read With Error, they are all LOW.

* Does not apply to Write cycles.

Diagram C. Read-Without-Error or Partial-Word-Write Cycle Timing

SWITCHING WAVEFORMS (Cont'd.)

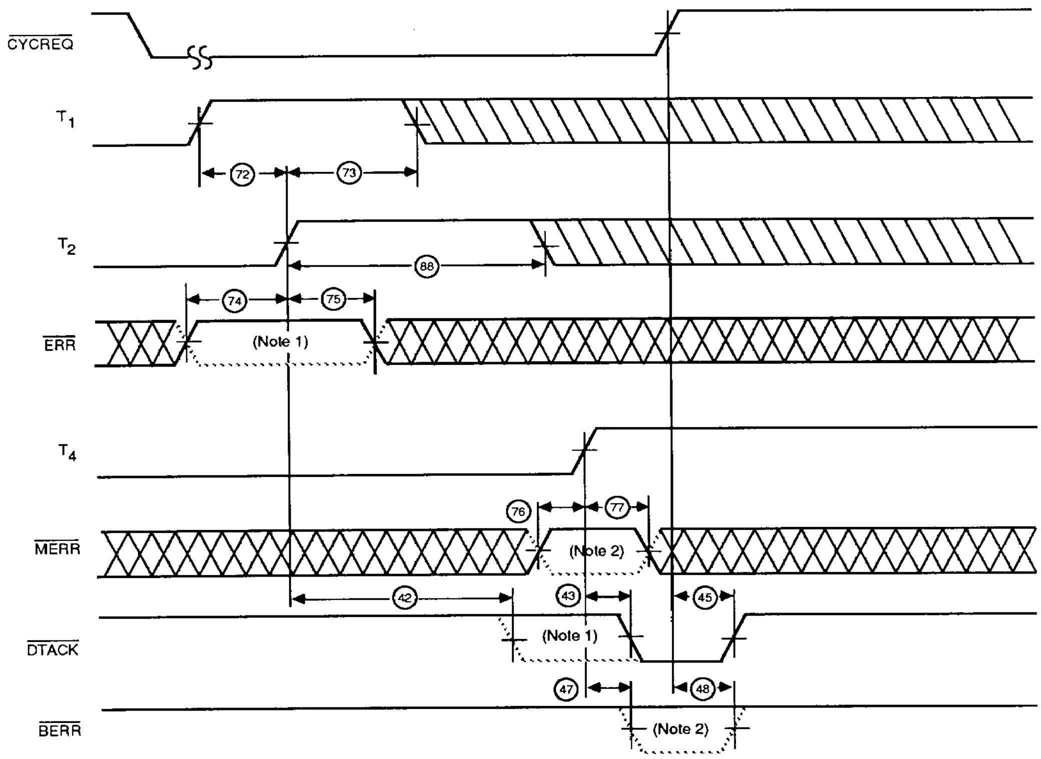


05221C-22

Notes: 1. When ERR is LOW, BERR and HALT will go LOW.

Diagram D. Bus-Retry Mode, First-Access Timing

SWITCHING WAVEFORMS (Cont'd.)



Notes: 1. When $\overline{ERR}$ is LOW, $\overline{DTACK}$ will be HIGH.
2. When $\overline{MERR}$ is LOW, $\overline{BERR}$ goes LOW.

05221C-23

Diagram E. Bus-Retry Mode, Second-Access Timing

SWITCHING WAVEFORMS (Cont'd.)

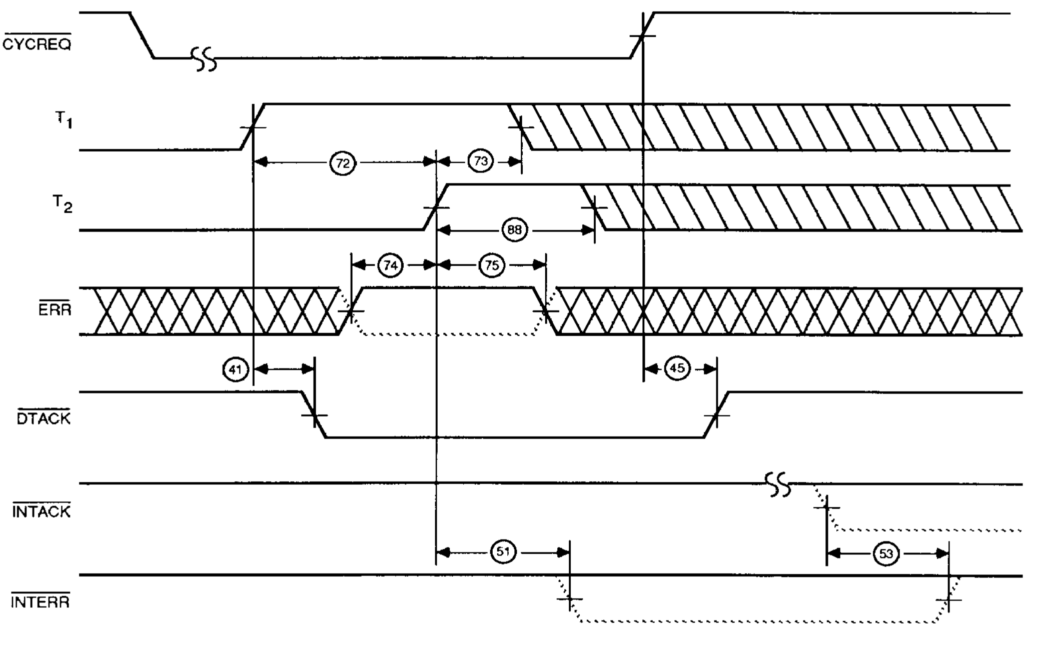
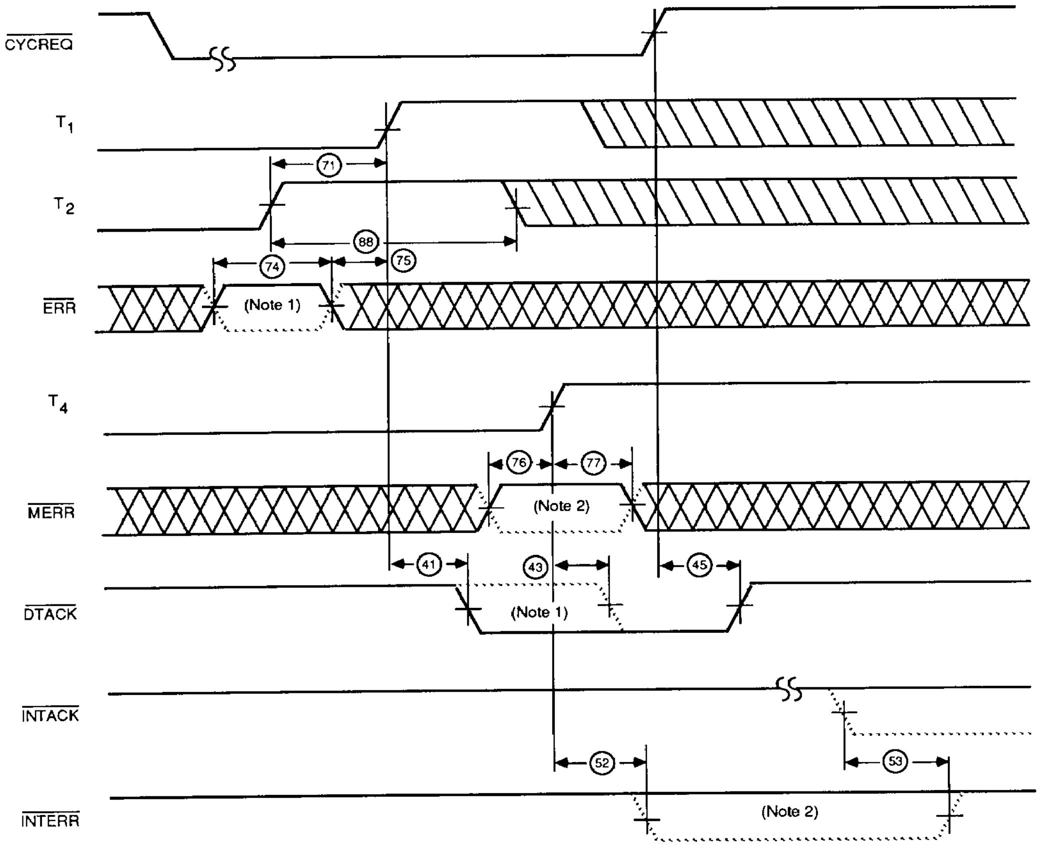

$\overline{\text{CYCREQ}}$
T_1
T_2
$\overline{\text{ERR}}$
$\overline{\text{DTACK}}$
$\overline{\text{INTACK}}$
$\overline{\text{INTERR}}$
72
73
88
74
75
41
45
51
53
05221C-24

Diagram F. Fly-By Mode Timing

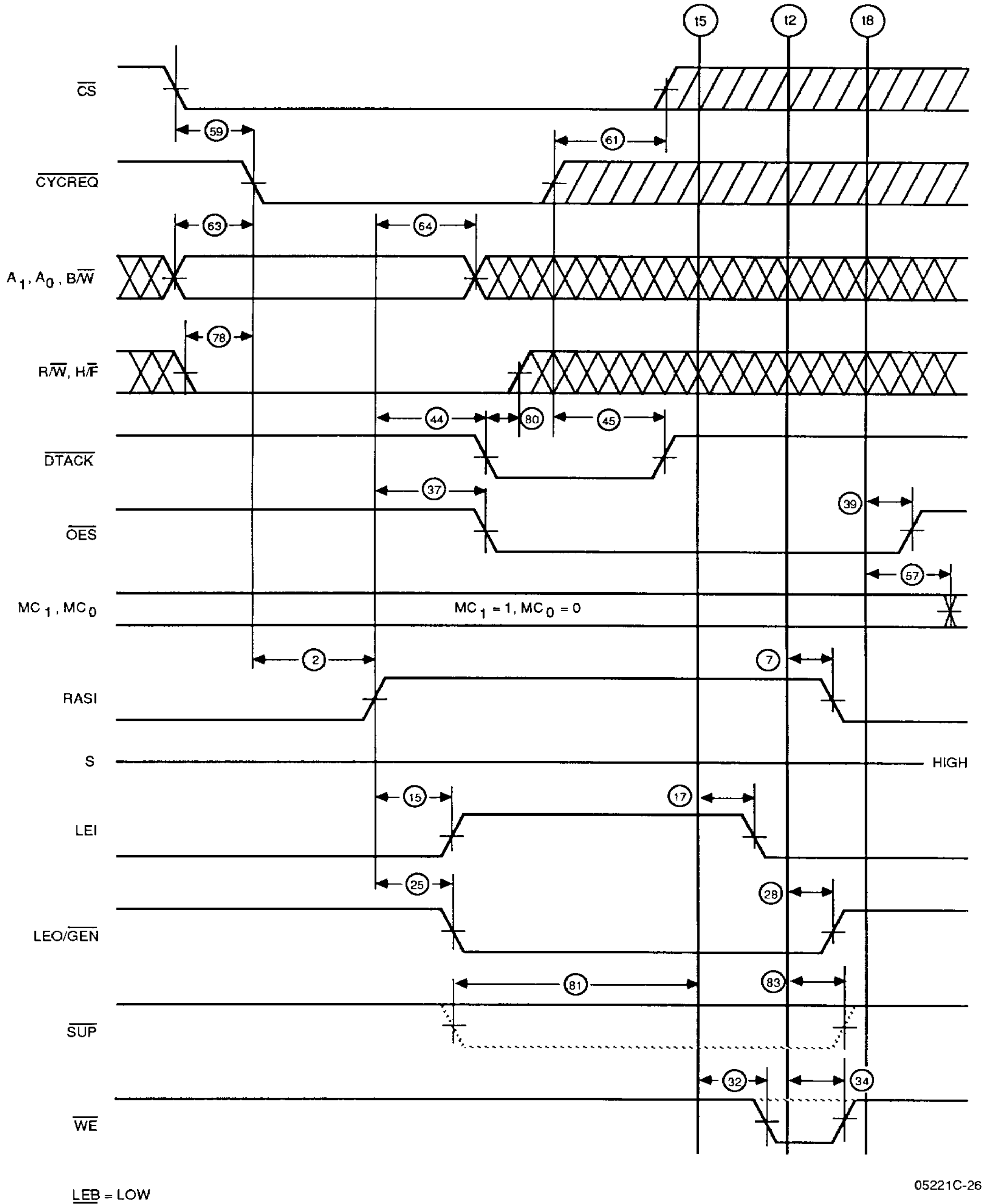


Notes: 1. When ERR is LOW, DTACK will be HIGH.
2. When MERR is LOW, INTERR goes LOW.

05221C-25

Diagram G. Flow-Thru Mode Timing

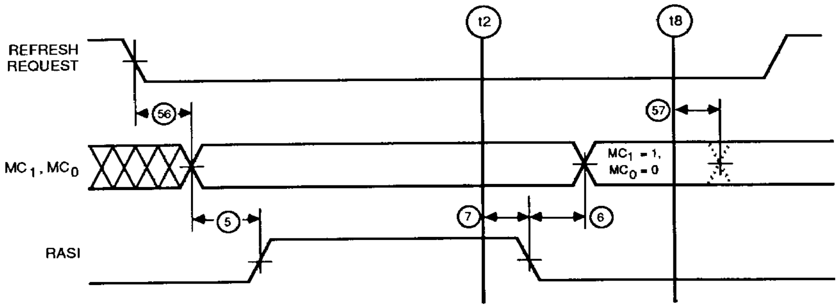
SWITCHING WAVEFORMS (Cont'd.)



$\overline{\mathrm{LEB}}$ = LOW
$\overline{\mathrm{OEL}_n}$ = 1
$\overline{\mathrm{OEH}_n}$ = 1
$\overline{\mathrm{OEBH}}$, $\overline{\mathrm{OEBL}}$ = 1

Diagram H. Full-Word-Write Cycle Timing

SWITCHING WAVEFORMS (Cont'd.)

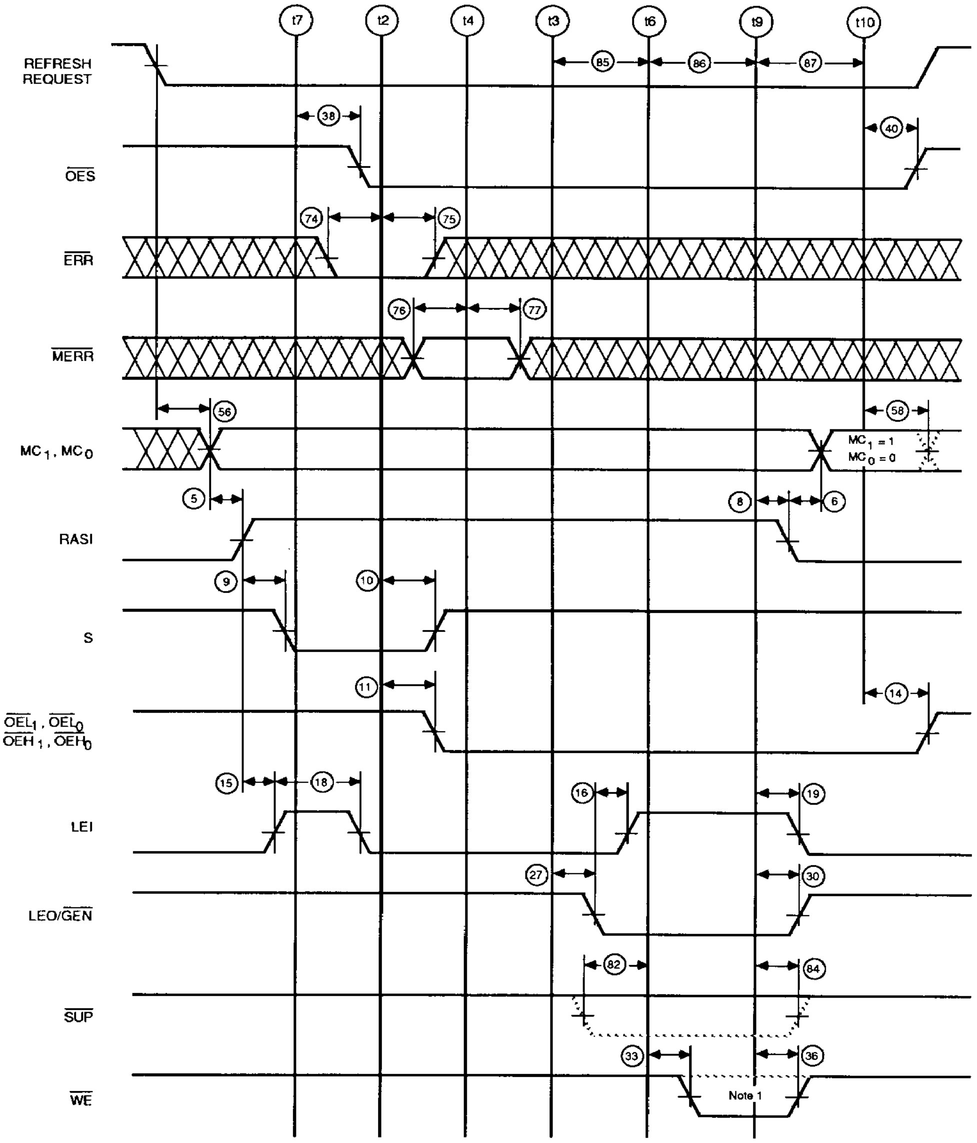


Notes: Refresh Request = $\overline{FR}$ with $\overline{CS}$ = X
OSC = $\overline{CS}$ = X (positive edge causes Refresh Request)

05221C-27

Diagram I. Refresh Cycle Timing

SWITCHING WAVEFORMS (Cont'd.)



Notes: Refresh Request = $\overline{FR}$ with $\overline{CS}$ = X
OSC with $\overline{CS}$ = X (positive edge causes Refresh Request)

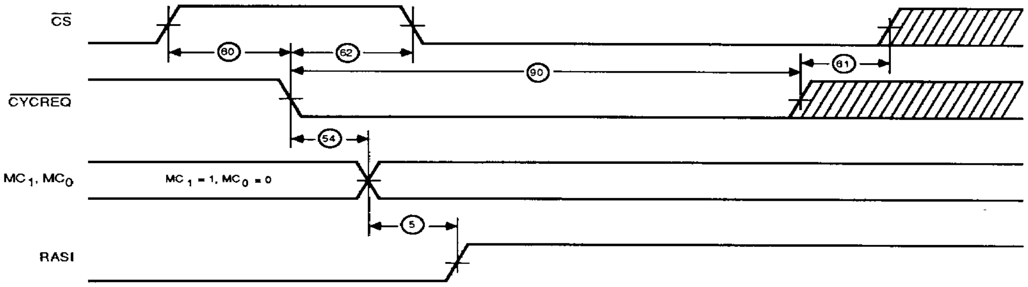
1. $\overline{WE}$ will remain HIGH if SUP is asserted.

LEB = LOW
$\overline{DTACK}$ = HIGH
$\overline{OEBH}$, $\overline{OEBL}$ = HIGH

05221C-28

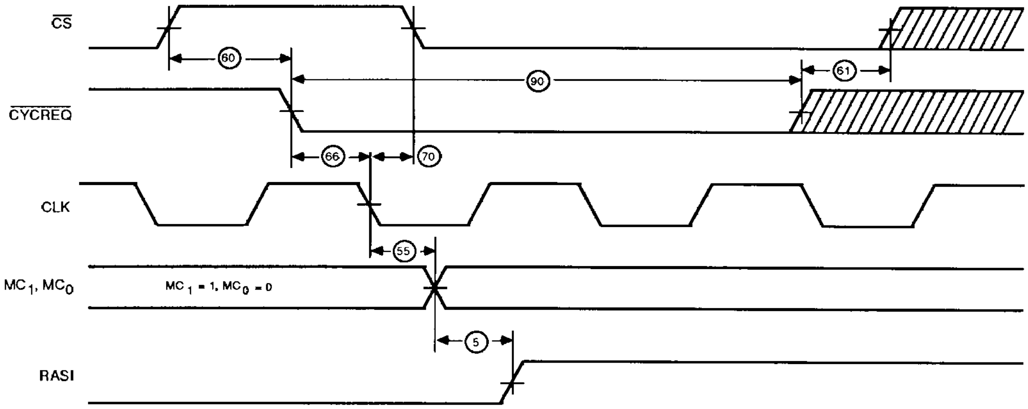
Diagram J. Refresh-With-Scrub Cycle Timing

SWITCHING WAVEFORMS (Cont'd.)



05221C-29

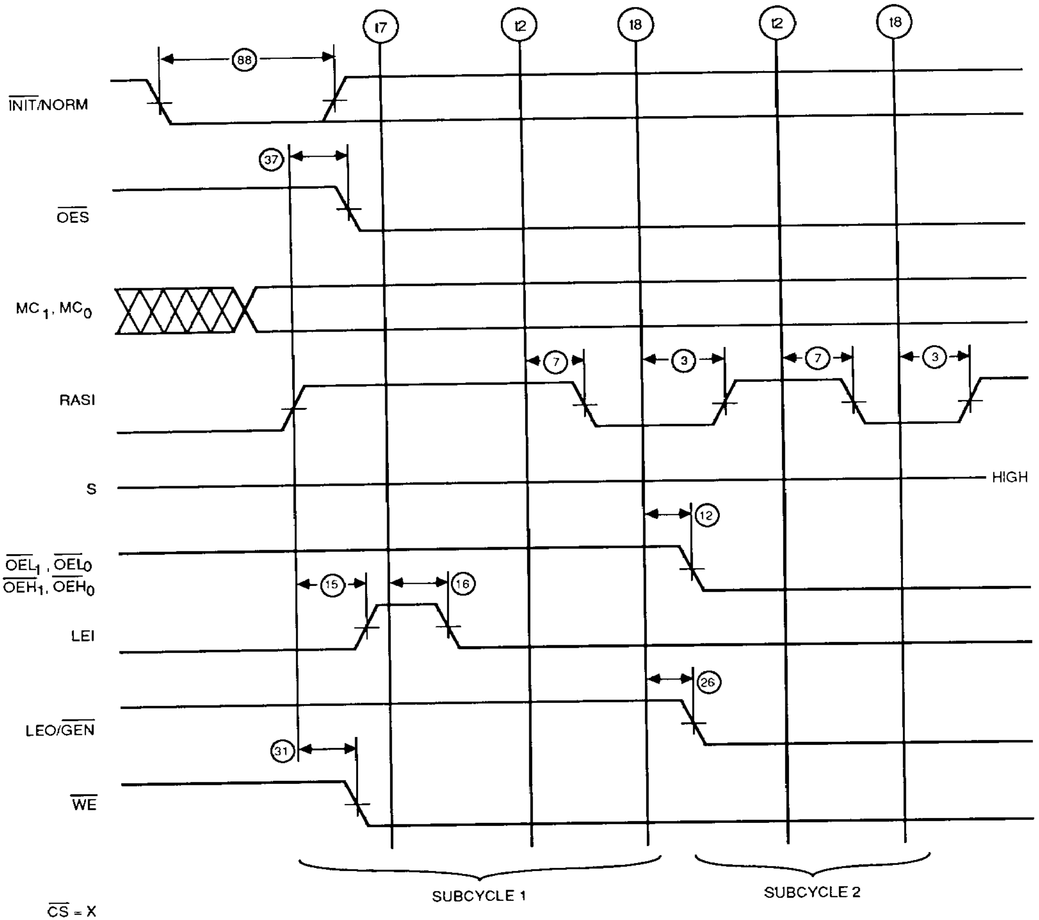
Diagram K. Hidden-Refresh Cycle Timing (Asynchronous Mode)



05221C-30

Diagram L. Hidden-Refresh Cycle Timing (Synchronous Mode)

SWITCHING WAVEFORMS (Cont'd.)

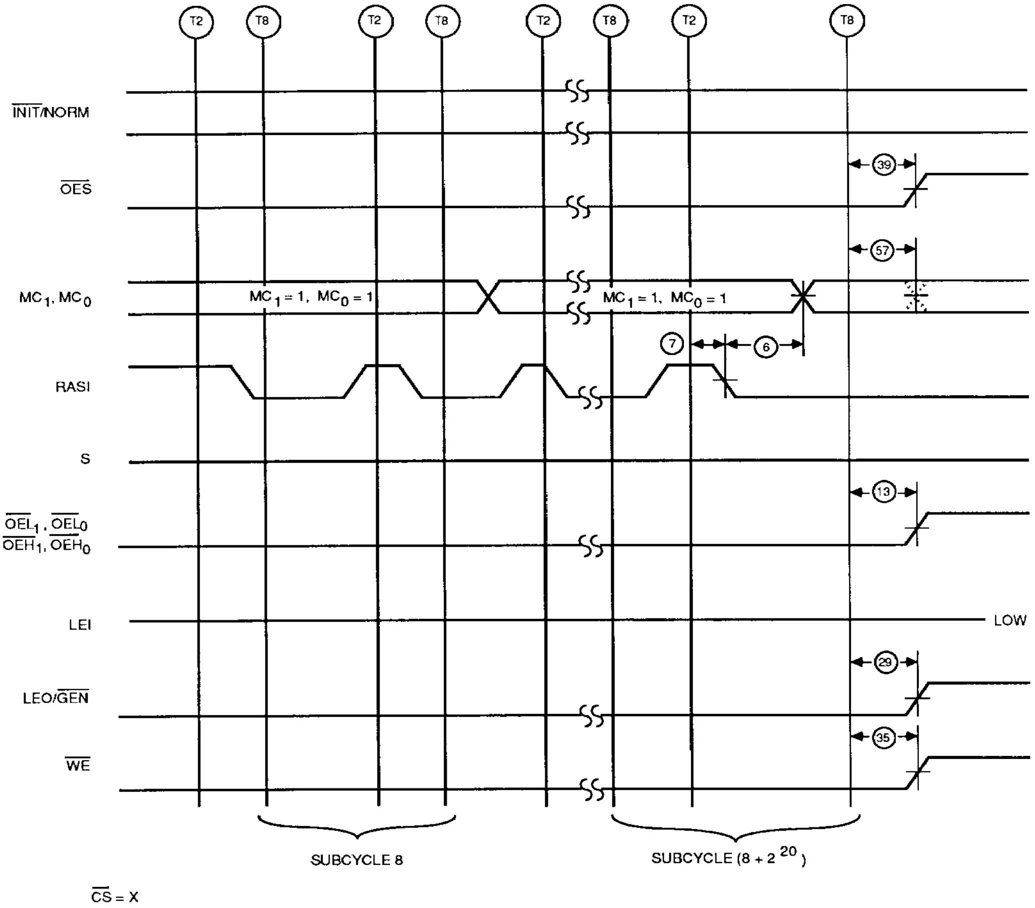


CS = X
DTACK = HIGH
LEB = LOW
OEBH, OEBL = HIGH

05221C-31

Diagram M. Initialize Cycle Timing

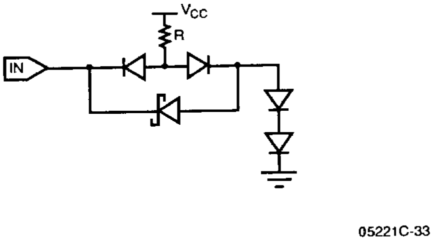
SWITCHING WAVEFORMS (Cont'd.)



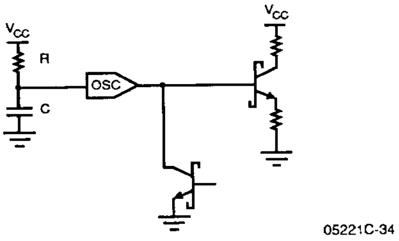
05221C-32

Diagram N. Initialize Cycle Timing (Cont'd.)

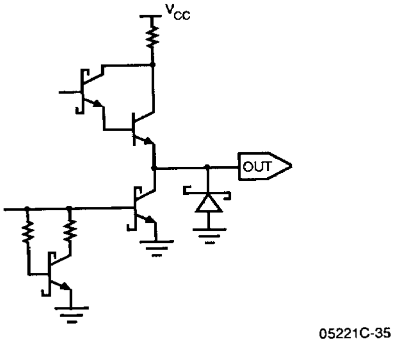
OUTPUT/INPUT CIRCUIT DIAGRAMS



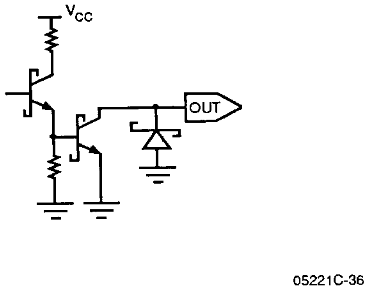
Typical Input
R = 17.6 kΩ
Note: SELECT Input has R = 12.8 kΩ



OSC Input
(R = 47 kΩ,
C = 1000 pF)



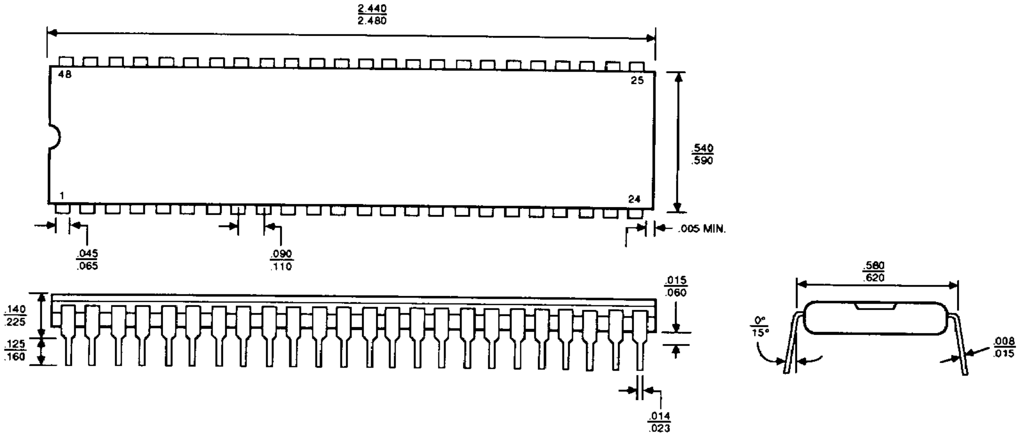
Typical Output



Open-Collector Outputs
(INTERR/HALT,
INTACK/BERR,
DTACK)

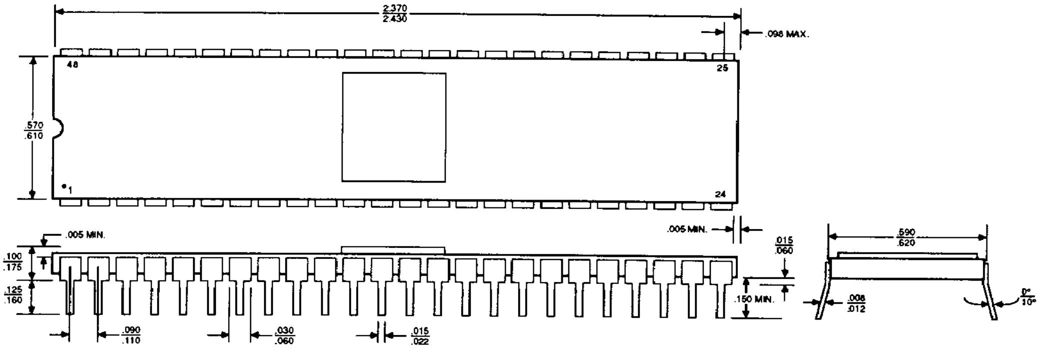
PHYSICAL DIMENSIONS*

PD 048



PID# 06566B

SD 048

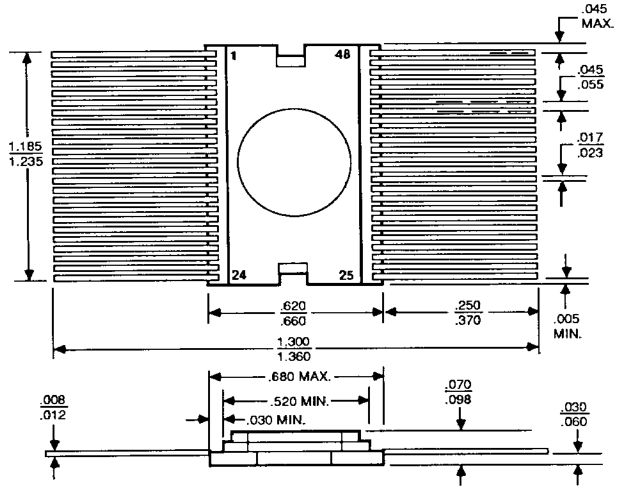


PID# 07546B

*For reference only

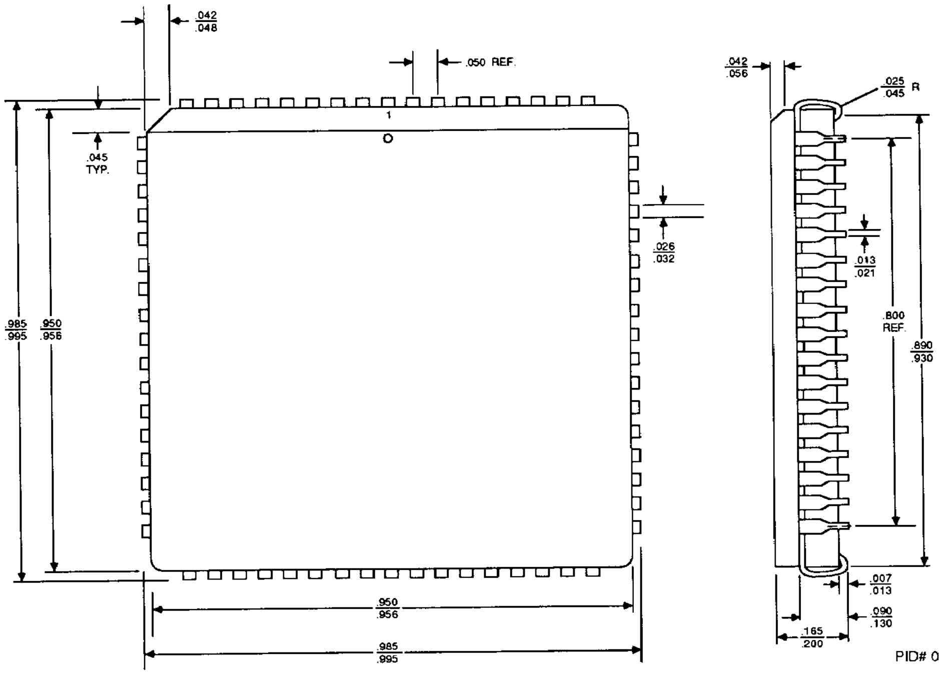
PHYSICAL DIMENSIONS (Cont'd.)

CFT 048



PID# 07869C

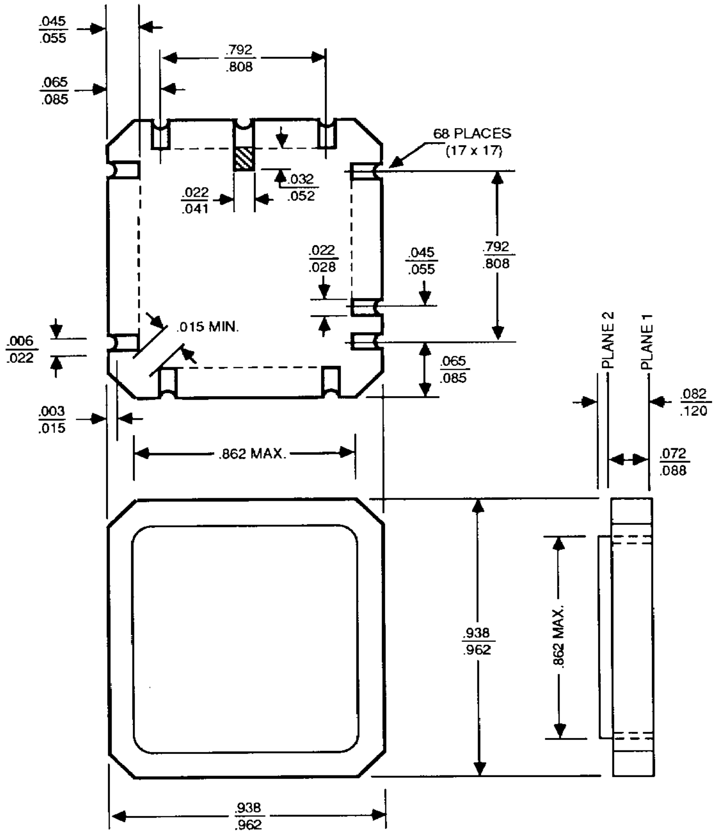
PL 068



PID# 067531

PHYSICAL DIMENSIONS (Cont'd.)

CL 068



PID# 07877A

ADVANCED MICRO DEVICES' NORTH AMERICAN SALES OFFICES

ALABAMA	(205) 882-9122
ARIZONA	(602) 242-4400
CALIFORNIA,	
Culver City	(213) 645-1524
Newport Beach	(714) 752-6262
San Diego	(619) 560-7030
Santa Clara	(408) 727-3270
Woodland Hills	(818) 992-4155
CANADA, Ontario,	
Kanata	(613) 592-0060
Willowdale	(416) 224-5193
FAX	(416) 224-5193
COLORADO	(303) 741-2900
CONNECTICUT	(203) 264-7800
FLORIDA,	
Clearwater	(813) 530-9971
Ft Lauderdale	(305) 484-8600
Melbourne	(305) 729-0496
Orlando	(305) 859-0831
GEORGIA	(404) 449-7920
ILLINOIS,	
Chicago	(312) 773-4422
Naperville	(312) 505-9517

INDIANA	(317) 244-7207
KANSAS	(913) 451-3115
MARYLAND	(301) 796-9310
MASSACHUSETTS	(617) 273-3970
MINNESOTA	(612) 938-0001
MISSOURI	(314) 275-4415
NEW JERSEY	(201) 299-0002
NEW YORK,	
Liverpool	(315) 457-5400
Poughkeepsie	(914) 471-8180
Woodbury	(516) 364-8020
NORTH CAROLINA	(919) 847-8471
OHIO	(614) 891-6455
OREGON	(503) 245-0080
PENNSYLVANIA,	
Allentown	(215) 398-8006
Willow Grove	(215) 657-3101
TEXAS,	
Austin	(512) 346-7830
Dallas	(214) 934-9099
Houston	(713) 785-9001
WASHINGTON	(206) 455-3600
WISCONSIN	(414) 792-0590

ADVANCED MICRO DEVICES' INTERNATIONAL SALES OFFICES

BELGIUM,		
Bruxelles	TEL	(02) 771 91 42
	FAX	(02) 762 37 12
	TLX	61028
FRANCE,		
Paris	TEL	(01) 49-75-10-10
	FAX	(01) 49-75-10-13
	TLX	263282F
GERMANY,		
Hannover area	TEL	(05143) 50 55
	FAX	(05143) 55 53
	TLX	925287
München	TEL	(089) 41 14-0
	FAX	(089) 406490
	TLX	523883
Stuttgart	TEL	(0711) 62 33 77
	FAX	(0711) 625187
	TLX	721882
HONG KONG,		
Kowloon	TEL	852-3-695377
	FAX	1234276
	TLX	504260AMDAPHX
ITALY, Milano	TEL	(02) 3390541
	FAX	(02) 3498000
	TLX	315286

JAPAN,		
Tokyo	TEL	(03) 345-8241
	FAX	3425196
	TLX	J24064AMDTKOJ
Osaka	TEL	06-243-3250
	FAX	06-243-3253
KOREA, Seoul	TEL	82-2-784-7598
	FAX	82-2-784-8014
LATIN AMERICA,		
Ft. Lauderdale	TEL	(305) 484-8600
	FAX	(305) 485-9736
	TLX	5109554261 AMDFTL
SWEDEN, Stockholm	TEL	(08) 733 03 50
	FAX	(08) 733 22 85
	TLX	11602
TAIWAN	TLX	886-2-7122066
	FAX	886-2-7122017
UNITED KINGDOM,		
Manchester area	TEL	(0925) 828008
	FAX	(0925) 827693
	TLX	628524
London area	TEL	(04862) 22121
	FAX	(04862) 22179
	TLX	859103

NORTH AMERICAN REPRESENTATIVES

CALIFORNIA	
I^2 INC	OEM (408) 988-3400 DISTI (408) 498-6868
CANADA	
Calgary, Alberta	
VITEL ELECTRONICS	(403) 278-5833
Kanata, Ontario	
VITEL ELECTRONICS	(613) 592-0090
Mississauga, Ontario	
VITEL ELECTRONICS	(416) 676-9720
Quebec	
VITEL ELECTRONICS	(514) 636-5951
IDAHO	
INTERMOUNTAIN TECH MKGT	(208) 888-6071
INDIANA	
ELECTRONIC MARKETING CONSULTANTS	(317) 253-1668
IOWA	
LORENZ SALES	(319) 377-4666
KANSAS	
LORENZ SALES	(913) 384-6556

MICHIGAN	
SAI MARKETING CORP	(313) 750-1922
MISSOURI	
LORENZ SALES	(314) 997-4558
NEBRASKA	
LORENZ SALES	(402) 475-4660
NEW MEXICC	
THORSON DESERT STATES	(505) 293-8555
NEW YORK	
NYCOM, INC	(315) 437-8343
OHIO	
Columbus	
DOLFUSS ROOT & CO	(614) 885-4844
Dayton	
DOLFUSS ROOT & CO	(513) 433-6776
Strongsville	
DOLFUSS ROOT & CO	(216) 238-0300
PENNSYLVANIA	
DOLFUSS ROOT & CO	(412) 221-4420
UTAH	
R^2 MARKETING	(801) 595-0631

Advanced Micro Devices reserves the right to make changes in its product without notice in order to improve design or performance characteristics. The performance characteristics listed in this document are guaranteed by specific tests, guard banding, design and other practices common to the industry. For specific testing details, contact your local AMD sales representative. The company assumes no responsibility for the use of any circuits described herein.

ADVANCED MICRO DEVICES 901 Thompson Pl., P.O. Box 3453, Sunnyvale, CA 94088, USA
TEL: (408) 732-2400 ● TWX: 910-339-9280 ● TELEX: 34-6306 ● TOLL FREE: (800) 538-8450

Printed in U.S.A. **TD-WCP-20M-10/87-0**